



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE CIENCIAS

CREACIÓN Y ADMINISTRACIÓN DE
CARTERAS DE INVERSIÓN MEDIANTE
ALGORITMOS GENÉTICOS.

T E S I S

QUE PARA OBTENER EL TÍTULO DE

LICENCIADO EN ACTUARÍA

P R E S E N T A :

DANIEL SÁNCHEZ RAMÍREZ.



FACULTAD DE CIENCIAS
UNAM

DIRECTOR DE TESIS :

M. EN C. JESÚS AGUSTÍN CANO GARCÉS.

2004



FACULTAD DE CIENCIAS
SECCION ESCOLAR



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ESTA TESIS NO SALE
DE LA BIBLIOTECA



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: Daniel Sánchez Ramírez

FECHA: 11 de octubre de 2004

FIRMA: [Firma]

ACT. MAURICIO AGUILAR GONZÁLEZ
Jefe de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo escrito:

“Creación y Administración de Carteras de Inversión mediante Algoritmos Genéticos”

realizado por Daniel Sánchez Ramírez

con número de cuenta 09712149-9 , quien cubrió los créditos de la carrera de: **Licenciatura en Actuaría.**

Dicho trabajo cuenta con nuestro voto aprobatorio.

A t e n t a m e n t e

Director de Tesis

Propietario

M. en C. Jesús Agustín Cano Garcés.

Propietario

Act. José Pablo Castañeda Martínez.

Propietario

M. en C. Jesús David Gómez Téllez.

Suplente

Act. Ana Lilia González López.

Suplente

Act. Gloria Roa Bejar.

Consejo Departamental de Matemáticas

[Firma]



Act. Jaime Vázquez Alamilla

Coordinador de la Carrera de Actuaría de CIENCIAS

CONSEJO DEPARTAMENTAL
DE
MATEMÁTICAS

CREACIÓN Y ADMINISTRACIÓN DE CARTERAS DE INVERSIÓN MEDIANTE ALGORITMOS GENÉTICOS

Daniel Sánchez Ramírez

Resumen

En esta tesis se tienen como objetos de estudio la creación y la administración de carteras de inversión mediante algoritmos genéticos. Con la intención de que el funcionamiento de los cinco algoritmos genéticos programados para realizar la creación y administración de carteras de inversión sea comprensible para el lector, previamente se realiza un estudio por separado de los dos temas que convergen en ellos: teoría de carteras de inversión y algoritmos genéticos. Posteriormente, se explica un bosquejo general de la forma en que fueron programados los cinco algoritmos genéticos y se les utiliza en dos ejemplos de carteras de inversión conformadas por acciones cotizantes en la Bolsa Mexicana de Valores; en el primero de ellos sólo se crea una cartera de inversión, mientras que en el segundo se crea y administra otra cartera de inversión. Finalmente, se concluye que los algoritmos genéticos programados son una buena opción para crear carteras de inversión, pero que necesitan ser mejorados para ofrecer también una buena alternativa para administrarlas.

Palabras clave: Carteras de inversión, acciones, algoritmos genéticos.

*Dedicado a cada una de las personas que han influido en mí
de forma positiva.*

AGRADECIMIENTOS

Deseo expresar mi más sincero agradecimiento a mi familia. A mis padres, Arturo y Guadalupe, y a mis hermanos, Leticia, Arturo y Jorge, quienes con su apoyo y orientación me han hecho el camino más fácil de andar. Les doy las gracias, porque en todo momento he percibido su firme intención de contribuir para hacer de mí un mejor individuo. También les agradezco su predisposición por generar un entorno que ha facilitado mi desarrollo intelectual, estimativo y como individuo.

Les agradezco a todos y cada uno de los profesores que han mostrado interés en mí aprendizaje y que con gran disposición por ayudarme, me han dado un minuto más de su tiempo cuando ha sido requerido. Agradezco en especial a aquellos profesores que han dedicado parte de su tiempo a la revisión de este trabajo, así como a quienes ayudaron a mejorarlo con sus valiosas sugerencias.

Por último, pero no por eso con menor sinceridad, les quiero agradecer a mis amigos por esos buenos momentos que hemos vivido y que, con el paso del tiempo, se convierten en excelentes pretextos para sonreír hasta en los momentos más difíciles.

INDICE

1. INTRODUCCIÓN	1
2. CARTERAS DE INVERSIÓN	11
2.1 Introducción a las carteras de inversión	11
2.2 Rendimiento y riesgo de un activo	15
2.2.1 Rendimiento de un activo	15
2.2.2 Rendimiento de un activo con ventas en corto	18
2.2.3 Riesgo de un activo	19
2.3 Rendimiento y riesgo de una cartera	20
2.3.1 Rendimiento de una cartera	20
2.3.2 Riesgo de una cartera	23
2.3.3 Diversificación	24
2.4 Optimización de carteras	27
2.4.1 Región factible	27
2.4.2 El conjunto de varianza mínima y la frontera eficiente	29
2.4.3 El modelo de Markowitz	31
2.5 Particularidades de la frontera eficiente	34
2.5.1 El teorema de dos fondos de inversión	34
2.5.2 Inclusión de un activo libre de riesgo	36
2.5.3 El teorema de un fondo de inversión	38
3. ALGORITMOS GENÉTICOS	40
3.1 Introducción a los algoritmos genéticos	40
3.2 Estructura de los algoritmos genéticos	44
3.2.1 Procedimiento de los sistemas evolutivos	44
3.2.2 Procedimiento de los algoritmos genéticos	45
3.2.3 Aplicación de los algoritmos genéticos en un problema sencillo	47
3.3 Operatividad de los algoritmos genéticos	51
3.3.1 Los algoritmos genéticos operando	51
3.3.2 Corrida de un algoritmo genético	54
3.3.3 Fundamentos de la operatividad de los algoritmos genéticos	61

4. ALGORITMOS GENÉTICOS PARA CARTERAS DE INVERSIÓN	68
4.1 Introducción	68
4.2 Elección de los activos financieros	69
4.2.1 Cálculos previos	70
4.2.2 Codificación de soluciones	71
4.2.3 Población inicial	73
4.2.4 Evaluación de la población	74
4.2.5 Selección de población	77
4.2.6 Cruzamiento	78
4.2.7 Mutación	80
4.2.8 Elitismo	81
4.3 Determinación de los pesos por asignar a los activos financieros	83
4.3.1 Cálculos previos	85
4.3.2 Codificación y factibilidad de soluciones	87
4.3.3 Población inicial	88
4.3.4 Evaluación de la población	89
4.3.5 Selección de población	91
4.3.6 Cruzamiento	92
4.3.7 Mutación	92
4.3.8 Elitismo	93
4.4 Determinación del punto de varianza mínima	93
4.4.1 Características generales del algoritmo genético 3	94
4.4.2 Evaluación de la población	95
4.5 Inclusión del activo libre de riesgo	96
4.5.1 Características generales del algoritmo genético 4	97
4.5.2 Evaluación de la población	98
4.6 Inclusión de la aversión al riesgo del inversionista	99
4.6.1 Características generales del algoritmo genético 5	100
4.6.2 Evaluación de la población	101
4.7 Administración de la cartera de inversión	102
5. EJECUCIÓN DE LOS ALGORITMOS GENÉTICOS	104
5.1 Introducción	104

5.2 Alteración de parámetros genéticos	106
5.2.1 Efectos en el algoritmo genético 1	107
5.2.2 Efectos en el algoritmo genético 2	112
5.2.3 Efectos en el algoritmo genético 5	117
5.3 Creación de una cartera de inversión con un universo de activos reducido	120
5.3.1 Selección de activos	120
5.3.2 Determinación de ponderaciones	122
5.3.3 Punto de varianza mínima	125
5.3.4 Inclusión del activo libre de riesgo	126
5.3.5 Inclusión del grado de aversión al riesgo del inversionista	128
5.4 Creación y administración de una cartera de inversión	130
5.4.1 Creación de la cartera de inversión	130
5.4.2 Administración de la cartera de inversión	133
5.4.3 Resultados obtenidos	136
5.4.4 Comportamiento del mercado	137
6. CONCLUSIONES	140
NOTAS	148
APÉNDICE 1: CÓDIGOS DE LOS ALGORITMOS GENÉTICOS	151
APÉNDICE 2: TABLAS DE RESULTADOS (PARÁMETROS GENÉTICOS)	173
BIBLIOGRAFÍA	187

1. INTRODUCCIÓN

Resulta muy agradable constatar como el ser humano, a lo largo de toda su historia, se ha preocupado por estudiar diversas áreas del conocimiento. Debido a éste interés mostrado en todas las épocas, en unas más que en otras, se ha logrado acumular una gran cantidad de información de cada una de las distintas materias. No obstante, la conclusión más contundente que siempre se ha obtenido, es que en realidad se sabe muy poco. A pesar de que esta conclusión parece indicar un rotundo fracaso, no es así.

Es muy común que esta diversidad de estudios en las distintas áreas de conocimiento, sea la fuente de inspiración para resolver nuevas problemáticas. La situación más común que se presenta de este aspecto, se da cuando en una área de conocimiento se tiene un problema que ha sido resuelto con las herramientas y mecanismos propios de la materia, pero que las soluciones que se obtienen no son tan satisfactorias como se quisiera. En estos casos, gracias al conocimiento de algunas personas estudiosas de diversas materias o algunas veces a la imaginación de otras, lo que se hace es tomar un mecanismo de solución que pertenece a una área distinta de donde se estaba trabajando, para posteriormente ser adaptado en términos de la primera materia a las exigencias que presenta el problema. Muchas veces esta forma de intentar resolver un problema resulta exitosa.

Esta tesis es una investigación que tiene como forma de proceder, la descrita en el párrafo anterior. El título de ella hace referencia al objetivo de estudio: *Creación y Administración de Carteras de Inversión mediante Algoritmos Genéticos*; la cual será una prueba más de esa forma de actuar que se ha comentado anteriormente. Como se puede observar, en el objetivo que se persigue en este trabajo convergen dos grandes temáticas. Por un lado, la creación y administración de carteras de inversión, la cual hace referencia a las ciencias de la administración, la economía y las matemáticas. Por otro lado, los

algoritmos genéticos, que son una rama importante de la inteligencia artificial, se remiten a las ciencias de la computación, la biología y las matemáticas. De esta forma, se tiene un problema muy importante de carácter financiero, que es resuelto mediante un procedimiento que no es propiamente financiero.

Para llevar a cabo la misión planteada, sin dejar de lado ninguna de las dos temáticas, se realiza en principio una investigación bilateral e independiente que posteriormente sirve de base para la consolidación del propósito primordial del trabajo. Es decir, primero se exponen las dos temáticas por separado para que más tarde, ya que se tengan todos los elementos necesarios, se pueda realizar una integración que resulte satisfactoria.

La decisión de estudiar en un principio a cada temática por separado se debe principalmente a la intención de que el trabajo no sólo sirva como un texto de referencia para la creación y administración de carteras de inversión mediante algoritmos genéticos, sino que también se pretende que el texto sirva como un material de apoyo para aquellas personas que únicamente estén interesadas en alguna de las dos temáticas, ofreciendo de una forma adecuada todos aquellos aspectos relevantes de cada tema¹.

Los motivos que hacen tomar mayor fuerza a la aclaración previa, son la gran cantidad de literatura existente sobre el tema de carteras de inversión y la facilidad de adaptación de los algoritmos genéticos. En aras de ser más claro con respecto a este punto se deben resaltar ciertos aspectos.

Comenzando con los algoritmos genéticos, aunque aún no han sido definidos, por el momento, se puede decir que son procedimientos inspirados en la genética y la selección natural que permiten resolver problemas de optimización. Al ser catalogados como algoritmos, por la misma definición de la palabra, se entienden como una serie de pasos a

realizar para resolver un problema, donde en cada paso se ejecutarán ciertas órdenes previamente establecidas. Una de las ventajas principales de estos algoritmos, es el hecho de que casi cualquier problema de optimización puede ser resuelto mediante ellos, bajo la única condición de que se sepa escribir al problema en términos de los algoritmos genéticos.

En el caso de las carteras de inversión, se cuenta con una amplia referencia bibliográfica donde se abordan todos los temas relacionados con la teoría de carteras. Entre lo más destacado para el trabajo que aquí se desarrolla, se podrían destacar los criterios de selección de activos para la formación de carteras de inversión y la formulación matemática del problema de optimización de carteras.

Con base en lo comentado en los dos párrafos anteriores, podría ser factible desarrollar el objetivo de estudio de este trabajo asumiendo esas características mencionadas de los algoritmos genéticos y de las carteras de inversión; es decir, considerando sólo esos conocimientos se podrían satisfacer de forma aceptable los cuestionamientos que más adelante se plantean con respecto al objetivo principal del trabajo. Con ello resultaría innecesaria toda la investigación que encierran los primeros capítulos del trabajo. Sin embargo, eso evitaría cumplir con una de las firmes intenciones de este, la de permitir a cualquier persona usarlo a pesar de que sólo tenga interés en alguna de las dos temáticas; y tampoco daría pie a que, en el mejor de los casos, el lector pudiera ser llevado por la inquietud de saber algo más, a la lectura del otro tema y de la forma en que ambos pueden ser integrados.

Una vez hecha esta aclaración, es importante hablar sobre las carteras de inversión y los algoritmos genéticos de forma introductoria, no obstante que en los siguientes capítulos se explicarán más profunda y formalmente. Esto se hace con el fin de

tener definidos los elementos básicos que aparecerán a lo largo de todo el desarrollo del presente.

El tema de Carteras de Inversión, muchas veces también llamado portafolios de inversión, se encuentra comprendido dentro de la teoría de carteras. Con el concepto de cartera de inversión se hace referencia a la combinación de valores, o activos financieros, que un inversor conjunta con fines de lucro. La forma de lucrar con la cartera de inversión es a través de la administración de los valores a lo largo de un periodo determinado de tiempo, intentando conciliar de la mejor forma posible los dos aspectos que son importantes para el inversor: la rentabilidad y el riesgo.

En el campo de la teoría de la formación de carteras, Harry Markowitz ocupa un lugar sumamente importante, con la elaboración de su modelo para la obtención de una cartera óptima. En este modelo se representa de forma explícita la conducta racional del inversor, que consiste en maximizar el rendimiento esperado y minimizar el riesgo. La cartera óptima es aquella que otorga la mejor combinación entre el rendimiento y el riesgo utilizando los activos del mercado. Aquí también es prudente considerar que la cartera óptima para una persona en particular, dependerá del nivel de aversión al riesgo que ésta posea.

Este modelo que fue formulado en 1952, cobró mayor fuerza con su publicación en el libro *Portfolio Selection: Efficient Diversification of Investments*, escrito por Markowitz en el año de 1959.

Los activos financieros que pueden representar a un potencial integrante de una cartera de inversión son muy diversos, por ejemplo las acciones, las divisas, las tasas de interés, los productos derivados, las opciones, etc. En este trabajo las carteras se encontrarán conformadas por acciones, por lo que se hablará sobre ellas y su mercado.

Una acción es una participación en el capital de una sociedad; representa el dinero que ponen los socios en el capital (o recursos propios de la empresa). El valor nominal de la acción es precisamente lo que los accionistas han puesto originalmente en los recursos propios de la empresa.

Para un inversionista, una acción es un activo financiero en el que se invierte con la esperanza de obtener rendimiento por medio de dividendos o por el aumento del precio de la acción. El precio de la acción dependerá de su cotización en el mercado, es decir en la bolsa, que a su vez depende de las expectativas de rentabilidad.

La bolsa es el mercado donde se negocian acciones. A ella acuden los inversores y los intermediarios financieros. El papel que desempeña la bolsa es propiciar un mercado ordenado de emisión de acciones donde las empresas puedan adquirir recursos y además facilitar a los inversores la posibilidad de comprar o vender acciones. Dentro de la economía de un país es relevante por las posibilidades de financiación que supone para la empresa, por la importancia del ahorro dentro de los individuos y por ser un “termómetro” de la economía.

El modelo de Markowitz, las acciones y la bolsa son los tres elementos, por el lado de las carteras de inversión, que mayor importancia juegan en la temática fundamental de este trabajo.

En lo que se refiere a los Algoritmos Genéticos, que también se les suele nombrar como GA (por sus siglas en inglés, *Genetic Algorithms*), la evolución y la computación se encuentran unidas.

Los algoritmos genéticos son *software*, procedimientos modelados a partir de la genética y la evolución. Están diseñados para hacer más eficiente la búsqueda de soluciones atractivas para problemas grandes y complejos. La búsqueda procede bajo la idea de la

supervivencia del más apto, es decir, se manipula gradualmente una población de soluciones potenciales del problema hasta que el más apto domina a la población.

Algunas veces, los investigadores se refieren a “algoritmos genéticos”, en otras, a “un algoritmo genético”, pero no a “el algoritmo genético” porque los algoritmos genéticos son una clase abundante de procedimientos relacionados que pueden tener muchos pasos distintos. Cada paso tiene diversas variaciones, por lo que la gama de procedimientos relacionados es amplia.

Todo esto comenzó a mediados del siglo XX, cuando varios biólogos experimentaron con las simulaciones en computadora de sistemas genéticos. A.S. Fraser, en particular, se comprometió en experimentos que parecían el inicio de los algoritmos genéticos.

Sin embargo, fue John Holland quien inició el desarrollo de ideas acerca de la teoría de sistemas adaptativos alrededor de 1965. El enseñó cursos de teoría de sistemas adaptativos en la Universidad de Michigan y comenzó a publicar varios textos sobre la materia. Gradualmente Holland refinó sus ideas, lo cual culminó con la realización de su afamado libro *Adaptation in Natural and Artificial Systems*, publicado en 1975. Holland es generalmente reconocido como el padre fundador de los algoritmos genéticos. Todos los pioneros en este campo tienen sus raíces en la Universidad de Michigan.

El punto de partida en el intento de usar algoritmos genéticos como una herramienta de solución es representar el problema de tal modo que el algoritmo pueda trabajar con él. Usualmente se utiliza una representación binaria, en la que a los parámetros de la solución del problema se les asigna una cadena de dígitos binarios, aunque no es la única.

De esta forma, lo primero que un algoritmo genético requiere es una buena representación de soluciones potenciales para poderlas operar; a esta representación se le conoce como codificación. Una vez que dicha representación se ha definido, comienza el proceso que está inspirado totalmente en la genética y la evolución.

Se pueden distinguir cinco pasos en los algoritmos genéticos. El primero es la generación aleatoria de un conjunto de soluciones posibles del problema. En términos de evolución, en este paso lo que se realiza es crear a la primera generación de la población.

En el segundo paso, utilizando una función previamente elaborada, se calcula un valor numérico que indique qué tan buen desempeño realiza cada una de las soluciones generadas en el paso 1. Esta función, conocida como de aptitud o desempeño, representa matemáticamente los criterios de optimalidad del problema; debido a esto, existen una o varias funciones de aptitud para cada problema. Su homólogo en el tema evolutivo, corresponde a la aptitud que tiene cada individuo de la generación para adaptarse a su medio ambiente.

El tercer paso consiste en la selección de las soluciones con mejor desempeño para que jueguen un papel de *padres* para procrear descendencia que conforme la segunda generación. Para realizar esta selección existen diversos métodos que ponderan de formas distintas el desempeño de cada solución para definir cuáles de ellas son las elegidas.

En el cuarto paso aparecen los operadores genéticos. Por medio de ellos se dará una nueva identidad a las nuevas soluciones, o *hijos*, provenientes de las soluciones de la primera generación seleccionadas para dejar descendencia. Los más importantes son el cruzamiento y la mutación que encuentran su referencia en su contraparte genética.

Por último, el quinto paso es repetir los pasos dos a cuatro con una nueva generación que ha sido creada, hasta que la población converja.

Ya que se han descrito de forma introductoria las dos temáticas, es momento de establecer los cuestionamientos de modo más concreto.

Esta tesis pretende entregar un procedimiento para crear y administrar carteras de inversión basado en los algoritmos genéticos. Como objetivo más general, se tiene la evaluación del nivel de éxito obtenido mediante estos algoritmos en un problema complejo de carácter financiero y con diversos criterios existentes como el que se ha mencionado. Para ello, se desea indagar acerca de la forma en que se ven afectados los resultados que otorga el modelo al variar los parámetros que definen la forma de operar de los algoritmos genéticos; como por ejemplo, el número de generaciones consideradas, los métodos de selección utilizados, las probabilidades de ocurrencia de los operadores genéticos, etc.

Como resultado de este trabajo, se desea entregar un conjunto de algoritmos genéticos que sea capaz de elegir un subconjunto de activos financieros, para crear una cartera de inversión y administrarla durante un periodo de tiempo determinado, considerando algunos aspectos como la diversificación de activos, el rendimiento que otorga el activo libre de riesgo y el grado de aversión al riesgo del inversionista.

La estructura que esta tesis sigue para intentar cumplir con esos objetivos es la siguiente. A continuación de este primer capítulo, el segundo aborda los temas más importantes sobre la teoría de carteras de inversión de modo teórico. Entre ellos se encuentran la teoría para el cálculo del rendimiento y riesgo de un activo y de una cartera; la optimización de carteras, donde se expone el modelo de Markowitz; y algunos teoremas con significado operacional sobre frontera eficiente. Muchos elementos expuestos en ese capítulo 2, serán retomados posteriormente para aplicarlos en la construcción de los algoritmos genéticos.

El capítulo 3 es la sección dedicada al estudio de los algoritmos genéticos. Ahí se desarrollan los principales componentes de éstos algoritmos de una forma más profunda y se muestran un par de ejemplos donde se optimizan funciones que no corresponden a algún problema financiero. Por último en este tercer capítulo, se destina una sección para explicar el trasfondo matemático que respalda la forma de operar de los algoritmos genéticos como una técnica de optimización; para ello, se analizan cada una de las partes que compone la estructura de los algoritmos genéticos hasta llegar al teorema del esquema.

En el capítulo 4, con base en la información que se ha plasmado en los dos capítulos previos, se procede a construir cinco algoritmos genéticos que permitirán realizar el proceso de creación y administración de una cartera de inversión. A lo largo del capítulo, se revisa el bosquejo general que permite la elaboración del código que ejecutará cada uno de los algoritmos genéticos. Por cada algoritmo, se indican los parámetros de entrada, los parámetros de salida y, con un grado importante de cuidado, se explican las codificaciones utilizadas y la forma de construcción de las funciones de aptitud.

En el capítulo 5, se realiza un análisis y evaluación de la creación y administración de carteras de inversión mediante el uso de algoritmos genéticos. Por esta razón, el capítulo puede visualizarse en dos partes.

En una primera parte, se analizan los efectos causados en la calidad de las soluciones entregadas por los algoritmos genéticos, al ser modificados los parámetros que definen las características de la corrida del algoritmo.

En la segunda parte, se procede a ejecutar los algoritmos genéticos para crear y administrar una cartera de inversión. En un primer ejemplo se considera un universo de activos disponible reducido para mostrar, algoritmo por algoritmo, los resultados que ofrece. En el segundo ejemplo, se considera un universo de activos disponibles con 54

acciones cotizantes en la Bolsa Mexicana de Valores y un activo libre de riesgo (Cetes a 28 días); con base en los resultados de los algoritmos genéticos, se crea una cartera de inversión y se le administra durante cuatro semanas². Posteriormente a la administración, se revisan los resultados obtenidos considerando además el comportamiento que presentó el mercado durante el periodo de administración de la cartera de inversión.

En la parte final de la tesis se presentan las conclusiones que se han podido obtener a partir de todo el trabajo desarrollado. En estas conclusiones se resaltan los principales hallazgos, es decir, aquellos aspectos positivos y negativos que dejó la aplicación de los algoritmos genéticos en la creación y administración de carteras de inversión. También se plantean algunas sugerencias que podrían ocasionar mejores resultados.

Por último se presenta la sección destinada a las notas aclaratorias, un apéndice en el que se muestran los códigos de los algoritmos genéticos programados y un apéndice con las tablas de resultados obtenidos en las ejecuciones de los algoritmos genéticos para analizar los efectos de la variación de parámetros.

Los algoritmos genéticos fueron desarrollados en el ambiente del programa "MATrix LABoratory"³ (MATLAB) en su versión 6.0. MATLAB es un programa de cálculo técnico y científico, que aunque en algunas aplicaciones es más lento que el código equivalente desarrollado en C/C++ ó Fortran, siempre es una herramienta de alto nivel para desarrollar aplicaciones técnicas y, debido a su forma sencilla de utilizar, puede aumentar la productividad de los programadores con respecto a otros entornos de desarrollo.

El equipo utilizado para ejecutar las corridas de los algoritmos genéticos programados tenía las características siguientes: AMD Athlon(tm) XP 2200+, con disco duro de 19 GB y 228 MB en RAM.

2. CARTERAS DE INVERSIÓN

2.1 Introducción a las carteras de inversión

Una cartera de inversión es un conjunto de activos financieros que son combinados en iguales o distintas proporciones; el objetivo de su creación es administrarla durante un determinado periodo de tiempo para obtener lucro. La elección de los activos financieros y las proporciones asignadas a cada uno de ellos dentro de la cartera de inversión, dependen de los objetivos y restricciones del inversionista.

Aunque no representa una obligación para el inversionista, existe un proceso general a seguir para formar una cartera de inversión. Su ejecución, ayuda a llevar a cabo una construcción y administración sistemática de la cartera de inversión que sea congruente con los objetivos del inversionista; además, propiciará la acumulación de información que servirá de base para evitar la necesidad de tomar decisiones arbitrarias, de pánico o euforia. Los pasos del proceso a seguir son: analizar las características, restricciones y perfil de riesgo del inversionista; declarar los objetivos de la cartera de inversión; definir una estrategia por ejecutar para administrar la cartera de inversión; realizar la selección estratégica de activos y de valores; y llevar a cabo la medición de resultados.

El primer paso es el análisis de las características, restricciones y perfil de riesgo del inversionista. En él se pretende delimitar la libertad de acción en la administración de la cartera de inversión y hacer explícitas las particularidades que condicionarán los objetivos del inversionista. Los dos resultados más importantes del análisis son la definición del tipo de inversionista y el nivel de riesgo aceptable. Además, se determinan características como el tamaño de la cartera de inversión; el horizonte temporal, es decir, el tiempo que permanecerán los activos en la cartera de inversión; y la rotación, o en otras palabras, el

número de veces permitido para vender y volver a comprar el total de los activos de la cartera de inversión al cabo del periodo de administración.

Como segundo paso se ubica la declaración de los objetivos de la cartera de inversión, que también puede ser considerada como el establecimiento de un conjunto de criterios para decidir la estrategia de administración y la composición de la cartera. Muchas veces estos objetivos se encuentran en contraposición unos con otros, especialmente en el caso de la rentabilidad y el riesgo: cuanto mayor rentabilidad se quiera, mayor riesgo se tendrá que asumir. Sin embargo, el objetivo más general será la optimización de la rentabilidad de acuerdo al nivel de riesgo aceptado.

Con la información obtenida en los dos primeros pasos, el perfil de riesgo – rentabilidad del inversionista es fácil de identificar y con base en él se determina, en algunos casos con toda precisión, qué estrategia de administración de carteras seguir. Para ello se debe, por un lado, recordar el origen y destino futuro del patrimonio y, por otro, considerar la cantidad monetaria máxima que el inversionista está dispuesto a perder como resultado de la posible ocurrencia de un escenario desfavorable.

Generalmente se identifican tres tipos de perfil de riesgo – rentabilidad: adverso al riesgo, indiferente al riesgo y propenso al riesgo. Un inversionista se dice ser adverso al riesgo si requiere un rendimiento más alto para correr un poco más de riesgo; en contra parte, un inversionista se dice ser propenso al riesgo si, con tal de obtener un rendimiento adicional, está dispuesto a tomar una cantidad mucho mayor de riesgo. Un inversionista es indiferente al riesgo si su comportamiento no se explica con base en el nivel de riesgo que aceptará, es decir, si no es ni adverso ni propenso al riesgo.

De acuerdo a su perfil, cada inversionista podrá elegir entre diferentes combinaciones de riesgo – rentabilidad. El riesgo y la rentabilidad, medidos por la

desviación estándar y el valor esperado de los rendimientos respectivamente (como posteriormente se explicará), forman combinaciones que suelen ser representadas mediante las llamadas curvas de indiferencia del inversionista. En la figura 2.1, se observa una posible curva de indiferencia de cada uno de los tres tipos de perfiles antes explicados.

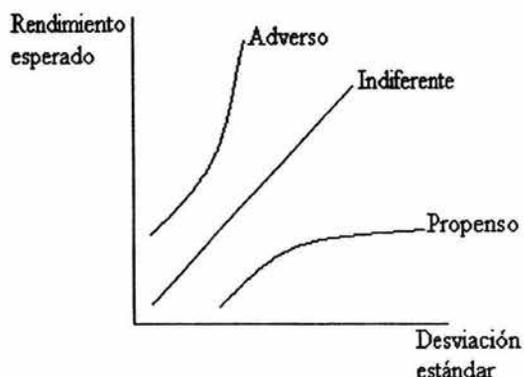


Figura 2.1

El tercer paso es definir la estrategia de administración a seguir; esta puede ser alguna de las técnicas que se han creado para gestionar carteras de inversión o una combinación de varias de ellas. La estrategia debe ser consistente con los objetivos previamente fijados. De forma general, las estrategias se pueden clasificar en pasivas y activas. En las estrategias pasivas se pretende que el comportamiento de la cartera de inversión, simule el comportamiento de un determinado índice o que no sobrepase un nivel máximo de riesgo fijo, sin importar el escenario que ocurra. Por el contrario, en las estrategias activas se pretende que la cartera de inversión, mediante el uso de diversas técnicas, supere un determinado índice objetivo. También se puede afirmar que las primeras, son propias de un perfil de inversionista conservador, con alta aversión al riesgo y que normalmente requieren poco gasto; mientras que las segundas, traen consigo costos mayores y, generalmente, son las sugeridas para alcanzar objetivos de crecimiento.

A la selección estratégica de activos y de valores, se le suele llamar distribución de activos. De modo más específico, la selección estratégica de activos es denominada distribución estratégica y la selección estratégica de valores es conocida como distribución táctica. La distribución de activos es el cuarto paso en la configuración de la cartera.

La distribución estratégica de los activos toma en cuenta los objetivos previamente fijados. En muchas ocasiones, los objetivos y las políticas del inversionista determinan el tipo de activos que deben ser incluidos en la cartera. El grado de riesgo de la cartera dependerá de los activos elegidos, del porcentaje invertido en cada tipo de activo y de la rotación de la cartera; en especial, éste último es un elemento que contribuye al mayor grado de especulación de la cartera. Algunas veces, se dice que la rentabilidad final de una cartera depende en un mayor grado de la distribución estratégica de activos que de la distribución táctica.

La distribución táctica cobra mayor importancia cuando un inversionista realiza el seguimiento de la evolución de los mercados y cuando es capaz de ubicar oportunidades puntuales de compra o venta de activos que puedan aprovecharse para aumentar el rendimiento de la cartera de inversión. Es por ello, que la composición básica de la cartera o distribución estratégica generalmente se mantiene, pero a través de la distribución táctica se pretende aprovechar las oportunidades que el mercado presenta a lo largo de su comportamiento en el tiempo. Las situaciones que comúnmente suelen aprovecharse mediante la distribución táctica son: la ubicación de los recorridos de algunos activos financieros a través de bandas de precios mínimos y máximos bien definidos; las ineficiencias del mercado, como pueden ser el efecto de enero o el de fin de año, meses en los que generalmente los precios de los activos suben; y las oportunidades puntuales para realizar operaciones de arbitraje sobre un activo financiero.

El último paso del proceso es la medición de resultados. Aquí se establece el sistema de medición que generará información sobre los rendimientos obtenidos en cada periodo de administración de la cartera, información sobre los activos de la cartera y sus dividendos, así como un resumen de las operaciones realizadas.

Como información complementaria, también se puede realizar un análisis de la situación del mercado y de las perspectivas para el siguiente periodo.

Al ejecutar este proceso, se pretende plantear una base sólida para lograr el mejor desempeño posible de la cartera de inversión. Sin embargo, jamás se puede solicitar sistemáticamente rendimientos por encima del mercado⁴.

2.2 Rendimiento y riesgo de un activo

2.2.1 Rendimiento de un activo

Existen diversas expresiones matemáticas para realizar el cálculo del rendimiento de un activo, sin embargo, la idea que sustenta a dichas expresiones siempre es la misma. El rendimiento de un activo indica la proporción entre la cantidad monetaria recibida como consecuencia de la posesión del activo y la cantidad monetaria pagada para obtenerlo. Si se utilizan R , P_0 , y P_1 para denotar respectivamente al rendimiento, el precio del activo al inicio del periodo y el precio del activo al final del periodo, se tiene que:

$$R = \frac{P_1}{P_0} \quad (2.1)$$

La fórmula (2.1) calcula el rendimiento simple del periodo y puede hacerse tan explícita como se desee. Por ejemplo, si se considera un activo financiero que otorga algún dividendo y se le denota por D , la expresión se modificaría de la siguiente forma:

$$R = \frac{P_1 + D}{P_0} \quad (2.2)$$

La tasa de rendimiento, denotada por r , es un concepto relacionado con el rendimiento. La expresión que da dicha relación es:

$$R = 1 + r \quad (2.3)$$

Sustituyendo la ecuación (2.1) en (2.3), se deduce que el valor de la tasa de rendimiento simple efectiva al plazo puede ser calculado como:

$$r = \frac{P_1 - P_0}{P_0} \quad (2.4)$$

Frecuentemente, se recomienda utilizar la tasa de rendimiento simple cuando se tienen datos anuales, ya sean históricos o estimados. Sin embargo, este cálculo de la rentabilidad presenta un inconveniente ya que sobrestima los rendimientos positivos y subestima los negativos; este sesgo se ve incrementado al anualizar los resultados.

Utilizando logaritmo natural se tiene otra técnica para calcular la tasa de rendimiento de un activo, esta es llamada tasa de rendimiento continuo⁵. Se recomienda el uso de esta técnica cuando se cuenta con datos mensuales o diarios; además, esta tasa de rendimiento no presenta el sesgo atribuido a la anterior. Su forma de cálculo es la siguiente:

$$r = \ln\left(\frac{P_1}{P_0}\right) \quad (2.5)$$

Es conveniente señalar que la tasa de rendimiento de los activos financieros suele manejarse a distintos plazos, pero es común que para efectos de comparación se utilicen tasas de rendimiento mensuales o anuales. La conversión de un plazo a otro puede calcularse mediante la siguiente fórmula:

$$r_n = (1 + r_p)^{n/p} - 1 \quad (2.6)$$

donde n representa el plazo al que se desea convertir la tasa de rendimiento que originalmente se calculó para un plazo p .

Tanto el rendimiento como la tasa de rendimiento, *a posteriori*, son magnitudes conocidas con certeza. No obstante, *a priori*, se trata de variables aleatorias que dependen de las expectativas. Cada una, como variable aleatoria, podrá tomar distintos valores de acuerdo a la función de densidad de probabilidad que la caracterice.

Una alternativa para estimar el rendimiento o la tasa de rendimiento de un activo, es considerar la esperanza matemática de la variable correspondiente. De esta forma, el rendimiento esperado de un activo financiero es calculado como:

$$E(R) = \sum_{i=1}^n Pr_i \cdot R_i \quad (2.7)$$

En la ecuación (2.7), Pr_i denota la probabilidad de que la variable aleatoria R adquiera el valor R_i .

En el caso de la tasa de rendimiento esperada, la expresión matemática que estima dicha variable es:

$$E(r) = \sum_{i=1}^n Pr_i \cdot r_i \quad (2.8)$$

Otro modo de calcular el rendimiento esperado de un activo financiero, es calcular la media de la serie de rendimientos históricos. En este caso se considera que la mejor estimación sobre el rendimiento futuro es el rendimiento histórico⁶. La expresión matemática para efectuar este cálculo es:

$$\bar{R} = \frac{\sum_{i=1}^n R_i}{n} \quad (2.9)$$

De igual forma, se puede suponer que la mejor estimación sobre la tasa de rendimiento futuro está dada por la media de la serie de tasas de rendimientos históricas, por lo que se tendría la siguiente expresión:

$$\bar{r} = \frac{\sum_{i=1}^n r_i}{n} \quad (2.10)$$

2.2.2 Rendimiento de un activo con ventas en corto

La venta en corto es una técnica usada por los inversionistas que intentan obtener alguna ganancia a partir de la caída del precio de un activo financiero. Una venta en corto es entendida como la venta de un activo financiero que no se posee.

Para hacer una venta en corto, primero el inversionista pide prestado el activo a un *broker*⁷ que lo posea. Inmediatamente después, el inversionista vende el activo a otra persona que esté interesada en poseerlo, de esa forma obtiene una cantidad P_0 . Después de un periodo de tiempo, el inversionista debe reponer el activo prestado mediante la compra de uno con características iguales al original, de este modo pagará una cantidad P_1 y podrá regresar el activo al corredor. Si el precio del activo estuvo a la baja, la cantidad P_1 será menor que la cantidad P_0 de tal forma que se obtiene un beneficio de $P_0 - P_1$. Entonces la venta en corto es conveniente si el precio del activo baja. En caso contrario, es decir, si la cantidad P_0 es menor que la cantidad P_1 , la pérdida es $P_1 - P_0$, pero P_1 puede incrementarse arbitrariamente, así como la pérdida.

La venta en corto es considerada bastante riesgosa por muchos inversionistas; la razón es justamente que si el valor del activo se incrementa, la pérdida potencial es ilimitada. Principalmente por esta razón, la venta en corto es prohibida dentro de algunas instituciones financieras y es evitada adrede como política por muchos individuos e instituciones. Inclusive en los mercados financieros, se establecen límites para el volumen máximo de ventas en corto permitidas y también se suele restringir su realización a mercados que tienen tendencias alcistas. A pesar de todo, las ventas en corto no son

universalmente prohibidas; de hecho, existe una proporción considerable de ventas en corto dentro de las operaciones realizadas en los mercados financieros.

Cuando se vende en corto un activo financiero, se debe considerar también, si el activo paga dividendos durante el periodo que se ha prestado. En caso de ser así, también se debe pagar el mismo dividendo a la persona que prestó el activo financiero.

Para calcular el rendimiento asociado con una venta en corto, se asume que en un principio se recibe un activo prestado por P_0 y después se paga P_1 por el activo, entonces el desembolso es $-P_0$ y el ingreso final es $-P_1$, por lo que el rendimiento total es:

$$R = \frac{-P_1}{-P_0} = \frac{P_1}{P_0}$$

Esta expresión es idéntica a la encontrada en la ecuación (2.1). Como antes, se puede deducir la tasa de rendimiento simple, para el caso en que las ventas en corto son permitidas, y concluir que:

$$r = \frac{P_1 - P_0}{P_0}$$

2.2.3 Riesgo de un activo

Al hablar de riesgo, se considera la posibilidad de obtener resultados extremos. Al medir el riesgo se desea saber qué tan grandes pueden ser esos resultados extremos y con qué probabilidad se darán. Es por eso que el riesgo de un activo financiero, también llamado volatilidad, se mide a través de la dispersión de sus posibles tasas de rendimiento.

En la mayoría de las ocasiones, la volatilidad de la tasa de rendimiento de un activo financiero es medida a partir de la desviación estándar. El cálculo de la desviación estándar sirve para cuantificar el riesgo, porque representa una medida de dispersión de las tasas de rendimiento con respecto a la media de las mismas.

De acuerdo a lo comentado antes, la volatilidad de las tasas de rendimiento de un activo financiero sería calculada como:

$$\sigma(r) = \sqrt{\sum_{i=1}^n \text{Pr}_i [r_i - E(r)]^2} \quad (2.11)$$

De la ecuación (2.11) se observa que la desviación estándar es calculada como la raíz cuadrada de la varianza.

La estimación histórica de la desviación estándar de las tasas de rendimiento de un activo, está dada por la expresión matemática siguiente:

$$\bar{\sigma}(r) = \sqrt{\frac{\sum_{i=1}^n (r_i - \bar{r})^2}{n-1}} \quad (2.12)$$

Cabe señalar que para estimar la volatilidad de las tasas de rendimiento de un activo financiero, existen más técnicas que pueden ser utilizadas. Algunas de ellas son la volatilidad dinámica (conocida también como volatilidad con suavizamiento exponencial) y la volatilidad implícita, las cuales no son abordadas en este trabajo.

2.3 Rendimiento y riesgo de una cartera

2.3.1 Rendimiento de una cartera

Partiendo del supuesto de la existencia de k diferentes activos que están disponibles para el inversionista, se puede crear una cartera de inversión compuesta por los n activos, con $1 \leq n \leq k$. Para ello también se supone que el inversionista es poseedor de una cantidad monetaria X , que está dispuesto a distribuir entre los n activos. Las cantidades seleccionadas X_i , con $i = 1, 2, \dots, n$, deben ser tales que $\sum_{i=1}^n X_i = X$, donde X_i representa la cantidad invertida en el i -ésimo activo.

Como se expuso antes, los inversionistas suelen realizar en ocasiones ventas en corto. Matemáticamente este hecho se vería reflejado en la posibilidad de que existan X_i con valores negativos; en contraparte, las restricciones que impiden a los X_i ser negativos se deben interpretar como la prohibición que se le hace al inversionista para realizar ventas en corto.

Con base en la idea de que la cartera de inversión es un sólo activo, las cantidades invertidas en cada uno de los activos originales se pueden considerar como fracciones de la inversión total. La relación matemática que refleja esta idea es la siguiente:

$$X_i = w_i X, \quad i = 1, 2, \dots, n$$

donde w_i es llamado el peso, fracción o ponderación del activo i en la cartera. Para ser congruente con la idea de la cartera de inversión como un sólo activo, los pesos deben ser tales que $\sum_{i=1}^n w_i = 1$ y nuevamente pueden existir w_i con valores negativos en caso de que el inversionista pueda realizar ventas en corto.

Denotando al rendimiento total del activo i , como R_i , la cantidad monetaria final generada en el periodo por el i -ésimo activo es $R_i X_i = R_i w_i X$. Entonces la cantidad total recibida por la cartera al final del periodo es:

$$\sum_{i=1}^n R_i w_i X$$

Ahora es posible calcular el rendimiento de la cartera completa mediante la expresión siguiente:

$$R = \frac{\sum_{i=1}^n R_i w_i X}{X} = \sum_{i=1}^n w_i R_i \quad (2.13)$$

De la expresión (2.13) se observa que el rendimiento de una cartera de inversión es igual a la suma de los correspondientes rendimientos de los activos individuales, ponderados por el peso relativo de cada activo con respecto de la cartera.

También de la ecuación (2.13), considerando que $R_i = 1 + r_i$ y que $\sum_{i=1}^n w_i = 1$, se desprende el resultado siguiente para la tasa de rendimiento de una cartera:

$$r = \sum_{i=1}^n w_i r_i \quad (2.14)$$

De la expresión (2.14) se tiene que la tasa de rendimiento de una cartera de inversión es igual a la suma de las correspondientes tasas de rendimiento de los activos individuales, ponderadas por el peso relativo de cada activo con respecto de la cartera.

Cuando se considera que las tasas de rendimiento de los activos son variables aleatorias, las cosas no se modifican significativamente⁸. Para observar esto, se suponen los n activos con tasas de rendimiento aleatorias r_1, r_2, \dots, r_n . Cada una tiene su valor esperado: $E(r_1) = \bar{r}_1; E(r_2) = \bar{r}_2; \dots; E(r_n) = \bar{r}_n$. Ahora bien, considerando la cartera formada por los n activos y usando los pesos $w_i, i = 1, 2, \dots, n$, se tendrá que la tasa de rendimiento de la cartera en términos de las tasas de rendimiento individuales es $r = w_1 r_1 + w_2 r_2 + \dots + w_n r_n$.

Calculando el valor esperado en ambos lados y apoyándose en la linealidad del valor esperado⁹, se tiene:

$$E(r) = w_1 E(r_1) + w_2 E(r_2) + \dots + w_n E(r_n) \quad (2.15)$$

A partir de la ecuación (2.15), es claro que siempre que se conozcan las tasas de rendimiento esperadas individuales y las ponderaciones asignadas a cada uno de los activos que integran la cartera de inversión, el cálculo de la tasa de rendimiento esperada puede ser encontrada de forma simple.

2.3.2 Riesgo de una cartera

Para calcular el riesgo de la cartera de inversión, es necesario obtener previamente una matriz de covarianzas; esto se debe a que la covarianza entre dos activos indica el comportamiento de un activo al producirse una variación en el valor del otro, además, como se observa a continuación, interviene en el cálculo que se pretende realizar. Para poder sustentar esta idea, considérese que la varianza del activo i se denota como σ_i^2 , la varianza del rendimiento de la cartera de inversión por σ^2 y la covarianza del rendimiento del activo i con el activo j por σ_{ij} . Entonces la varianza de la cartera es:

$$\begin{aligned}\sigma^2 &= E[(r - \bar{r})^2] \\ &= E\left[\left(\sum_{i=1}^n w_i r_i - \sum_{i=1}^n w_i \bar{r}_i\right)^2\right] \\ &= E\left[\left(\sum_{i=1}^n w_i (r_i - \bar{r}_i)\right)\left(\sum_{j=1}^n w_j (r_j - \bar{r}_j)\right)\right] \\ &= E\left[\sum_{i,j=1}^n w_i w_j (r_i - \bar{r}_i)(r_j - \bar{r}_j)\right] \\ &= \sum_{i,j=1}^n w_i w_j \sigma_{ij}\end{aligned}$$

Como resultado final se puede escribir:

$$\sigma^2 = \sum_{i,j=1}^n w_i w_j \sigma_{ij} \quad (2.16)$$

La ecuación (2.16) permite conocer el riesgo de la cartera de inversión. En ella se observa que la varianza de la tasa de rendimiento de una cartera puede ser calculada, de forma relativamente simple, a partir de las covarianzas existentes entre las series de las tasas de rendimientos de los activos y los pesos asignados a los activos para la formación de la cartera de inversión.

La covarianza indica en qué medida dos activos se mueven en el mismo sentido; si la covarianza es positiva, entonces cuando un activo sube el otro también; si la covarianza es negativa, quiere decir que cuando uno sube, el otro baja. Si la covarianza es próxima a cero, significa que los dos activos son independientes.

Otro parámetro estadístico que indica la relación entre dos activos, y que es más fácil de interpretar, es el coeficiente de correlación ρ . Este coeficiente viene dado por la expresión siguiente:

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \quad (2.17)$$

y siempre se cumple que $-1 \leq \rho \leq 1$. Utilizando la expresión (2.17), la ecuación (2.16) puede ser rescrita como:

$$\sigma^2 = \sum_{i,j=1}^n w_i w_j \rho_{ij} \sigma_i \sigma_j \quad (2.18)$$

La ventaja principal del coeficiente de correlación se encuentra en su rango acotado, ventaja que no puede atribuirse al cálculo de la covarianza.

2.3.3 Diversificación

Como se ha mencionado, el riesgo de una cartera de inversión está dado por la variabilidad de las tasas de rendimiento esperadas de los activos que la conforman. Entonces la intención del inversionista será disminuir esta variabilidad tanto como sea posible; es por ello que recurren al proceso conocido como diversificación.

En general, la varianza de la tasa de rendimiento de una cartera puede ser reducida mediante la inclusión de activos adicionales en ella, ya que las carteras constituidas con pocos activos pueden estar sujetas a un alto grado de riesgo. Este proceso refleja la idea de la frase que dice “no ponga todos los huevos en la misma canasta”.

La diversificación “puede” provocar la reducción de la variabilidad de la tasa de rendimiento esperado de la cartera debido a que los precios de los activos financieros no se mueven “perfectamente” unos con otros. Esto indica que en un mismo momento, mientras algunos activos se mueven a la alza, otros lo hacen a la baja y otros están estáticos¹⁰. Los resultados de esas fluctuaciones tienden a cancelarse unos a otros. En general, la variabilidad de la tasa de rendimiento de la cartera es menor que la variabilidad promedio de las tasas de rendimiento de sus componentes; esto debido a que la variabilidad de la cartera no sólo depende del riesgo inherente de sus activos, sino también de cómo esos activos se relacionan entre sí.

El riesgo que puede ser potencialmente eliminado por la diversificación es conocido como riesgo no sistemático, único o propio. El riesgo no sistemático resulta del hecho de que muchos de los peligros que rodean a una determinada empresa son específicamente suyos y “tal vez” de sus competidores inmediatos¹¹.

Pero hay también un riesgo que no se puede evitar por mucho que se diversifique. Este riesgo es conocido como riesgo sistemático, de mercado o no diversificable. El riesgo sistemático deriva del hecho de que hay otros peligros en el conjunto de la economía que amenazan a todos los negocios; por ejemplo, el crecimiento de una economía o la inflación. Por eso los activos tienden a moverse en el mismo sentido. Esta es la razón por la que los inversionistas están expuestos a las incertidumbres del mercado, independientemente del número de activos que incluyan en sus carteras.

Para ver el efecto de la diversificación en términos cuantitativos, considérese el caso siguiente. Supóngase que existen n activos y que la tasa de rendimiento de cada activo está no correlacionada con la de cualquier otro activo de los $n-1$ restantes. Además

supóngase que la tasa de rendimiento de cada uno de los activos tiene media μ y varianza σ^2 . Ahora se crea la cartera considerando proporciones iguales de los n activos; esto es, $w_i = 1/n$ para cada i .

Obsérvese que la tasa de rendimiento de la cartera es $r = \frac{1}{n} \sum_{i=1}^n r_i$, el valor de la media es $\bar{r} = \mu$, el cual es independiente de n y la varianza correspondiente es $\text{var}(r) = \frac{1}{n^2} \sum_{i=1}^n \sigma^2 = \frac{\sigma^2}{n}$, donde se usa el hecho de que las tasas de rendimiento individuales están mutuamente no correlacionadas. A partir de la expresión encontrada para la varianza, se puede ver que la varianza decrece mientras n crece; entonces la inclusión de un mayor número de activos provocará la disminución del riesgo de la cartera.

Ahora considérese el caso siguiente en el que las tasas de rendimiento de los activos están correlacionadas. En esta ocasión, nuevamente se supone que cada activo tiene una tasa de rendimiento con media μ y varianza σ^2 , pero ahora cada par de tasas de rendimiento tiene una covarianza de $\sigma_{ij} = .22\sigma^2$ para $i \neq j$ ¹². Si las proporciones de los n activos son iguales, el cálculo de la varianza de la cartera de inversión indica lo siguiente:

$$\begin{aligned} \text{var}(r) &= E \left[\sum_{i=1}^n \frac{1}{n} (r_i - \bar{r}) \right]^2 \\ &= \frac{1}{n^2} E \left\{ \left[\sum_{i=1}^n (r_i - \bar{r}) \right] \left[\sum_{j=1}^n (r_j - \bar{r}) \right] \right\} \\ &= \frac{1}{n^2} \sum_{i,j} \sigma_{ij} \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{n^2} \left[\sum_{i=j} \sigma_{ij} + \sum_{i \neq j} \sigma_{ij} \right] \\
&= \frac{1}{n^2} \left[n\sigma^2 + .22(n^2 - n)\sigma^2 \right] \\
&= \frac{.78\sigma^2}{n} + .22\sigma^2
\end{aligned}$$

En este caso la varianza jamás será menor de $.22\sigma^2$. Este análisis de diversificación, aunque hace demasiados supuestos que difícilmente se presentarán en un mismo momento en un mercado financiero, es capaz de mostrar de forma cuantitativa los efectos de la diversificación.

Así, la diversificación puede reducir el riesgo de una cartera, en el mejor de los casos, hasta el valor definido completamente por el riesgo sistemático. Gráficamente, el efecto de la diversificación sobre el riesgo de una cartera adopta la forma mostrada en la figura 2.2.

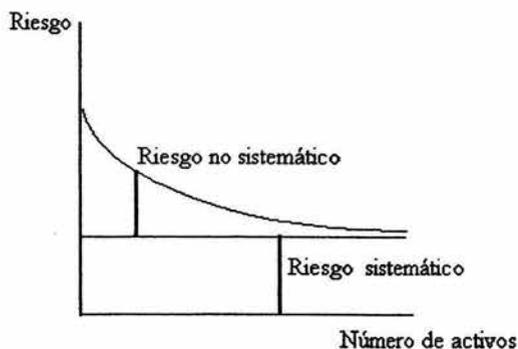


Figura 2.2

2.4 Optimización de carteras

2.4.1 Región factible

Nuevamente supóngase que se puede disponer de k activos para formar una cartera de inversión. A partir de esos k activos, se pueden construir carteras constituidas por

n activos, donde $1 \leq n \leq k$. El número de carteras posibles es demasiado grande, ya que a partir de una cartera con determinados activos como componentes, se puede crear una infinidad de carteras con los mismos activos pero con distintos coeficientes de peso w_i tales

$$\text{que } \sum_{i=1}^n w_i = 1.$$

Ahora bien, de acuerdo a las fórmulas (2.15) y (2.16) se puede calcular la tasa de rendimiento esperada y la desviación estándar de cada una de las carteras posibles. De este forma, cada cartera podrá ser representada por una pareja ordenada (*desviación estándar, rendimiento*). El conjunto de parejas ordenadas que representan a las carteras es conocido como conjunto factible o región factible. Esa región puede ser unidimensional, cuando sólo se consideran dos activos; pero cuando se tienen al menos tres activos no perfectamente correlacionados¹³, la región factible es una región sólida bidimensional.

Si se consideran, por ejemplo, las carteras posibles a partir de únicamente dos activos A y B , y se dibujan los puntos riesgo – rendimiento obtenidos sobre una gráfica, en la mayoría de las ocasiones se obtiene algo muy similar a lo mostrado en la figura 2.3.

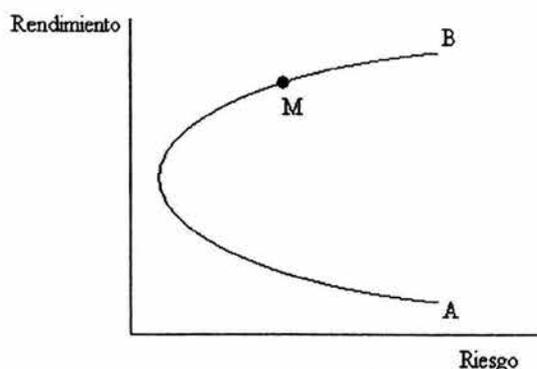


Figura 2.3

En la figura 2.3, cada punto de la región factible representa una cartera con una proporción distinta entre A y B . El extremo A representa una cartera invertida al cien por

ciento en el activo A , y el extremo de B representa el cien por ciento invertido en B . El punto M representa una cartera en la que $0 \leq w_A \leq 1$, $0 \leq w_B \leq 1$ y $w_A + w_B = 1$.

La forma de la región factible puede verse modificada de forma significativa como consecuencia de los valores que posean las varianzas de los activos involucrados y de sus covarianzas.

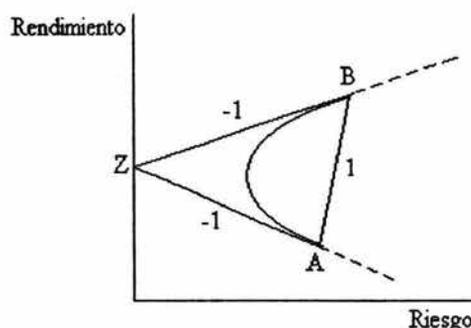


Figura 2.4

En la figura 2.4 se muestran las combinaciones posibles, a partir de dos activos, cuando el coeficiente de correlación tiene alguno de sus valores extremos. Se puede observar que las líneas ZA y ZB representan posibles combinaciones cuando el coeficiente de correlación vale -1 ; mientras que la línea AB representa las posibles combinaciones cuando el coeficiente de correlación vale 1 .

Una vez que se ha construido la región factible, se enfrenta una pregunta obvia, ¿qué cartera es la más conveniente? En otras palabras, se desea saber que criterio utilizar para realizar la elección adecuada de aquella cartera que otorgue los mejores resultados al inversionista.

2.4.2 El conjunto de varianza mínima y la frontera eficiente

El conjunto de varianza mínima está formado por los puntos factibles que, para un rendimiento dado, tienen menor varianza que cualquier otro punto de la región factible.

Con base en la forma de construir la región factible, se puede deducir que para cualquier valor de la tasa de rendimiento esperada, el punto factible con la menor varianza (o desviación estándar) es el correspondiente al punto límite izquierdo de dicha región.

En la figura 2.5 se observa la forma característica de un conjunto de varianza mínima. Dentro de éste conjunto se puede ubicar a un punto especial; conocido como el punto de varianza mínima (PVM), él representa a la cartera de inversión con menor riesgo.

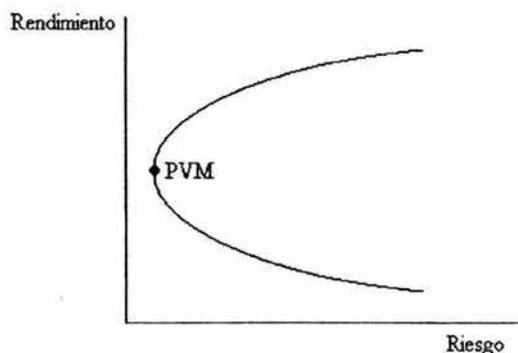


Figura 2.5

Se supone que los inversores rehuyen el riesgo, es decir, entre dos activos con rendimientos iguales y riesgo diferente, el inversionista siempre preferirá el activo con menor riesgo. A esta idea se le conoce como el criterio de varianza mínima: entre dos activos de igual rendimiento esperado, se preferirá aquel que tiene menor riesgo.

A veces también se suele utilizar otro criterio para decidir entre una cartera y otra, que puede ser visto como el caso complementario del criterio de varianza mínima. Se le conoce como el criterio del inversionista insaciable y señala que un inversionista elegirá la cartera con mayor rendimiento esperado para un nivel de riesgo dado.

Ambos criterios implican que para los inversionistas que son adversos al riesgo e insaciables, solamente es importante la "parte superior" del conjunto de varianza mínima. A esta región se le llama frontera eficiente de la región factible.

En la figura 2.6 se muestra una frontera eficiente, que es entendida como el conjunto de portafolios eficientes, en el sentido de que otorgan las mejores combinaciones de riesgo – rendimiento para la mayoría de los inversionistas.

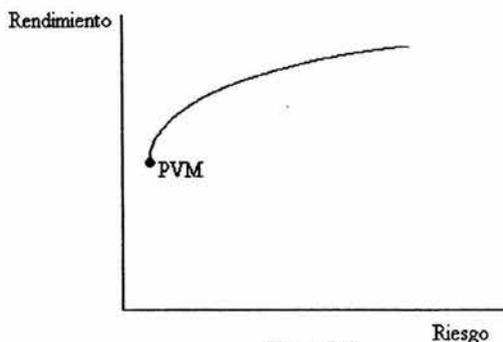


Figura 2.6

2.4.3 El modelo de Markowitz

Harry Markowitz fue el primero en hablar de carteras eficientes para referirse a aquellas que aumentaban el rendimiento esperado para un nivel de riesgo dado. El problema de encontrar estas carteras eficientes, consiste en asignar el importe limitado de capital para alcanzar el rendimiento más alto dado un nivel de riesgo.

Para formular un problema matemático que guíe a las carteras de varianza mínima, se asume que hay n activos cuyas tasas de rendimiento son $\bar{r}_1, \bar{r}_2, \dots, \bar{r}_n$ y las covarianzas son σ_{ij} , para $i, j = 1, 2, \dots, n$; además se toma en cuenta que una cartera es definida como un conjunto de n pesos $w_i, i = 1, 2, \dots, n$, que suman 1. Para encontrar una cartera de varianza mínima, se fija el valor esperado en un valor deseado para \bar{r} . Entonces se busca la cartera factible de varianza mínima que tenga esta tasa de rendimiento. El problema es formulado como sigue:

$$\text{Minimizar } \sum_{i,j=1}^n w_i w_j \sigma_{ij} \quad (2.19)$$

$$\text{sujeto a } \sum_{i=1}^n w_i \bar{r}_i = \bar{r} \quad (2.20)$$

$$\sum_{i=1}^n w_i = 1 \quad (2.21)$$

El problema de Markowitz es la base de la teoría de inversión para un periodo. A partir de éste, se ubica de forma clara el intercambio entre la tasa de rendimiento esperada y la varianza de la tasa de rendimiento esperada de una cartera. Una vez formulado el problema de Markowitz, se puede resolver numéricamente para obtener una solución específica.

Por ejemplo, la técnica de los multiplicadores de Lagrange λ y μ , permite encontrar la solución del problema de Markowitz. Para ello es necesario en un principio, calcular el Lagrangiano, derivarlo con respecto a cada variable e igualar cada expresión a cero. El Lagrangiano en este caso es:

$$L = \sum_{i,j=1}^n w_i w_j \sigma_{ij} - \lambda \left(\sum_{i=1}^n w_i \bar{r}_i - \bar{r} \right) - \mu \left(\sum_{i=1}^n w_i - 1 \right)$$

Considerando el caso de dos variables se tiene que:

$$L = (w_1^2 \sigma_1^2 + w_1 w_2 \sigma_{12} + w_2 w_1 \sigma_{21} + w_2^2 \sigma_2^2) - \lambda (\bar{r}_1 w_1 + \bar{r}_2 w_2 - \bar{r}) - \mu (w_1 + w_2 - 1)$$

Entonces

$$\frac{\partial L}{\partial w_1} = (2w_1 \sigma_1^2 + w_2 \sigma_{12} + w_2 \sigma_{21}) - \lambda \bar{r}_1 - \mu$$

$$\frac{\partial L}{\partial w_2} = (w_1 \sigma_{12} + w_1 \sigma_{21} + 2w_2 \sigma_2^2) - \lambda \bar{r}_2 - \mu$$

Usando el hecho de que $\sigma_{12} = \sigma_{21}$ e igualando las derivadas a cero, se tiene que:

$$2\sigma_1^2 w_1 + 2\sigma_{12} w_2 - \lambda \bar{r}_1 - \mu = 0$$

$$2\sigma_{21} w_1 + 2\sigma_2^2 w_2 - \lambda \bar{r}_2 - \mu = 0$$

De esta forma se obtienen dos ecuaciones, que junto a las dos de las restricciones, forman un sistema de cuatro. Este sistema puede ser resuelto, por medio de técnicas de álgebra lineal, para las cuatro incógnitas w_1 , w_2 , λ y μ .

El procedimiento se puede extender fácilmente a n variables. Entonces los n pesos w_i de la cartera, con $i = 1, 2, \dots, n$, y los dos multiplicadores de Lagrange λ y μ para una cartera eficiente que tiene tasa de rendimiento esperada \bar{r} satisfacen:

$$\sum_{j=1}^n 2\sigma_{ij}w_j - \lambda\bar{r}_i - \mu = 0, \text{ para } i = 1, 2, \dots, n. \quad (2.22)$$

$$\sum_{i=1}^n w_i\bar{r}_i = \bar{r} \quad (2.23)$$

$$\sum_{i=1}^n w_i = 1 \quad (2.24)$$

De esta forma, se tienen un total de $n + 2$ ecuaciones con $n + 2$ incógnitas. La solución de estas ecuaciones, que además son lineales, producirá los pesos para la cartera eficiente con la tasa de rendimiento esperada dada.

Hasta el momento, los signos de las variables w_i no fueron restringidos por lo que el problema deja abierta la posibilidad de realizar ventas en corto. Cuando se desea prohibir las ventas en corto, es necesario incluir las restricciones que impiden a las w_i ser negativas. El planteamiento alternativo del problema de Markowitz queda escrito como:

$$\text{Minimizar } \sum_{i,j=1}^n w_i w_j \sigma_{ij} \quad (2.25)$$

$$\text{Sujeto a } \sum_{i=1}^n w_i \bar{r}_i = \bar{r} \quad (2.26)$$

$$\sum_{i=1}^n w_i = 1 \quad (2.27)$$

$$w_i \geq 0, \text{ para } i = 1, 2, \dots, n \quad (2.28)$$

Este es llamado un programa cuadrático, ya que el objetivo es cuadrático y las restricciones son ecuaciones o desigualdades lineales. El inconveniente grave que presenta es que no puede ser reducido para la solución de un conjunto ecuaciones lineales.

Existe una diferencia marcada entre las dos formulaciones explicadas que es fácilmente observable en los valores óptimos w_i . Cuando las ventas en corto son permitidas, la mayoría de los valores óptimos w_i tienen valores distintos de cero (positivos ó negativos). En contraste, cuando las ventas en corto no son permitidas, se hace uso de unos cuantos activos ya que generalmente muchos pesos óptimos son igual a cero.

El desarrollo de todos éstos principios básicos de la formación de carteras, lo realizó Harry Markowitz, culminándolo con su publicación en un artículo en 1952. Hoy en día, el modelo de Markowitz sigue siendo la piedra angular en la teoría de carteras.

2.5 Particularidades de la frontera eficiente

2.5.1 El teorema de dos fondos de inversión

Previamente se ha desarrollado la técnica de los multiplicadores de Lagrange para dar solución al problema de Markowitz. Resolviendo el problema para cada rendimiento posible de esa forma, se encontraría el conjunto de varianza mínima, es decir, los puntos en este conjunto satisfacen el siguiente sistema de $n + 2$ ecuaciones lineales:

$$\sum_{j=1}^n 2\sigma_{ij} w_j - \lambda \bar{r}_i - \mu = 0, \text{ para } i = 1, 2, \dots, n. \quad (2.29)$$

$$\sum_{i=1}^n w_i \bar{r}_i = \bar{r} \quad (2.30)$$

$$\sum_{i=1}^n w_i = 1 \quad (2.31)$$

Si se supone que el problema ha sido resuelto en un par de ocasiones, para las tasas de rendimiento esperadas \bar{r}^1 y \bar{r}^2 , entonces se tendrían dos soluciones distintas $w^1 = (w_1^1, w_2^1, \dots, w_n^1)$, λ^1 , μ^1 y $w^2 = (w_1^2, w_2^2, \dots, w_n^2)$, λ^2 , μ^2 para el sistema de ecuaciones.

A partir de las dos soluciones encontradas, se pueden hacer combinaciones mediante la multiplicación de la primera por α y la segunda por $(1-\alpha)$, con $-\infty < \alpha < \infty$. Las combinaciones de esta forma también representan una solución a las $n + 2$ ecuaciones, correspondiente un valor esperado de $\alpha\bar{r}^1 + (1-\alpha)\bar{r}^2$; en consecuencia, se puede generar el conjunto de varianza mínima a partir de dos soluciones del problema de Markowitz. De la misma forma, a partir de dos soluciones originalmente eficientes, se puede construir la frontera eficiente. La forma de resaltar este resultado es como se ve en el siguiente párrafo.

Teorema de los dos fondos de inversión: Pueden establecerse dos fondos de inversión eficientes para que cualquier fondo eficiente pueda reproducirse, en términos de su media y su varianza, como una combinación de estos dos. En otras palabras, todos los inversionistas que buscan fondos eficientes necesitan solamente invertir en combinaciones de estos dos fondos.

Para comprobar la validez de este resultado, debe observarse que $\alpha w^1 + (1-\alpha)w^2$ es una cartera factible ya que sus pesos suman 1, con lo que (2.31) es satisfecha. También se tiene que el rendimiento esperado es $\alpha\bar{r}^1 + (1-\alpha)\bar{r}^2$, con lo que se satisface (2.30). Finalmente, para notar que (2.29) se cumple, debe recordarse que las dos soluciones consideradas hacían igual a cero el lado izquierdo esa restricción; entonces su combinación también lo hace. Con (2.29), (2.30) y (2.31) satisfechas, la cartera que resulta de la

combinación $\alpha w^1 + (1-\alpha)w^2$ es también una solución al problema; es decir, también representa un punto en el conjunto de varianzas mínimas.

Este resultado tiene importantes aplicaciones. Una de ellas es una implicación computacional. Para resolver (2.24) para todos los valores de \bar{r} , ahora solamente es necesario encontrar dos soluciones y después formar combinaciones de ellas.

Otra de las implicaciones puede ser ubicada en el ámbito operacional. Un fondo mutuo es una alternativa de inversión que se construye con las aportaciones de inversionistas individuales que tienen objetivos de inversión similares y que en vez de invertir en forma particular, delegan la administración de su dinero a una empresa especializada. Entonces de acuerdo al teorema de los dos fondos de inversión, a partir de dos fondos mutuos se puede prestar un servicio completo de inversión para todas las personas.

2.5.2 Inclusión de un activo libre de riesgo

Los inversionistas tienen la oportunidad de prestar o pedir prestado a la tasa libre de riesgo. Cuando se habla de prestar a la tasa libre de riesgo, como puede ser en la compra de un bono, el activo libre de riesgo adquiere un peso positivo; por el contrario, pedir prestado corresponde a tener un peso negativo.

Esta posibilidad provoca que, en el análisis de la teoría de selección de carteras, se considere un activo libre de riesgo en la lista de los activos disponibles. Un activo libre de riesgo es un instrumento financiero cuyo rendimiento es cierto; es por esta razón que se puede afirmar que su varianzas es cero y, en consecuencia, su desviación estándar es nula.

La inclusión de un activo libre de riesgo induce una modificación en la forma de la frontera eficiente. Para observar el efecto de la inclusión del activo libre de riesgo,

supóngase que éste otorga una tasa de rendimiento r_f y que existe otro activo con riesgo A que tiene tasa de rendimiento r_a , con media \bar{r}_a y varianza σ_a^2 . A partir de estos dos activos el inversionista formará su cartera.

La forma de combinar estos dos activos será usando un peso de α para el activo libre de riesgo y $1 - \alpha$ para el activo con riesgo, con $\alpha \leq 1$. Entonces el rendimiento esperado de la cartera es $\alpha r_f + (1 - \alpha) \bar{r}_a$.

Antes de calcular el riesgo asociado a esta cartera, es conveniente hacer notar que la covarianza entre los rendimientos de los dos activos considerados es cero. Esto se debe a que la diferencia entre el rendimiento del activo libre de riesgo y el valor esperado del rendimiento del activo libre de riesgo es cero, entonces $E[(r_a - \bar{r}_a)(r_f - r_f)] = 0$.

La desviación estándar del rendimiento de la cartera puede ser calculado como $\sigma = \sqrt{\alpha^2 \sigma_f^2 + 2\alpha(1 - \alpha)\sigma_{fa} + (1 - \alpha)^2 \sigma_a^2} = (1 - \alpha)\sigma_a$. En el cálculo anterior se hace uso del hecho de que la covarianza entre los dos activos y la varianza del activo libre de riesgo es cero. Dado que $\sigma_f = 0$, también se puede escribir $\sigma = \alpha\sigma_f + (1 - \alpha)\sigma_a$.

A partir de las expresiones encontradas para el cálculo del rendimiento esperado y de la desviación estándar asociados a la cartera, las parejas ordenadas que representarán a las carteras factibles son de la forma $(\alpha\sigma_f + (1 - \alpha)\sigma_a, \alpha r_f + (1 - \alpha)\bar{r}_a)$. Se puede observar que ambas coordenadas dependen linealmente de α ; como consecuencia, cuando α varía, el punto que representa la cartera traza una línea recta en el plano varianza - rendimiento.

Con base en lo explicado en el párrafo anterior, cuando se tienen dos o más activos con riesgo y uno sin riesgo, la forma de la región factible es visualizada como un triángulo infinito. En la figura 2.7 se observa la degeneración de la región factible.

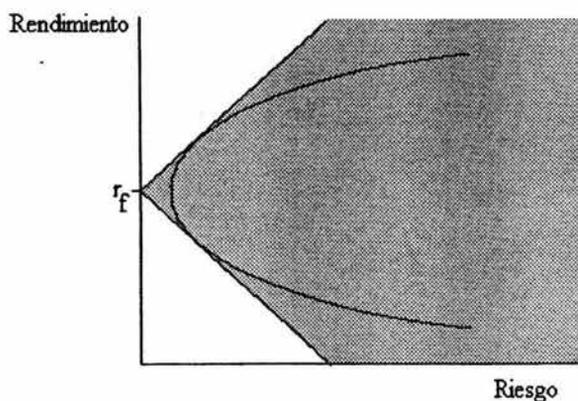


Figura 2.7

2.5.3 El teorema de un fondo de inversión

Cuando es posible incluir un activo libre de riesgo en el universo de los activos disponibles para el inversionista, la forma de la frontera eficiente también se ve afectada. Nuevamente con base en los criterios de varianza mínima y del inversionista insaciable, se puede deducir que en este caso la frontera eficiente consiste únicamente de la línea recta superior del triángulo de la región factible. Esta línea es tangente al conjunto original factible de activos con riesgo. Como se observa en la figura 2.8, existe un punto F en el conjunto original factible que está sobre la línea que define la frontera eficiente nueva.

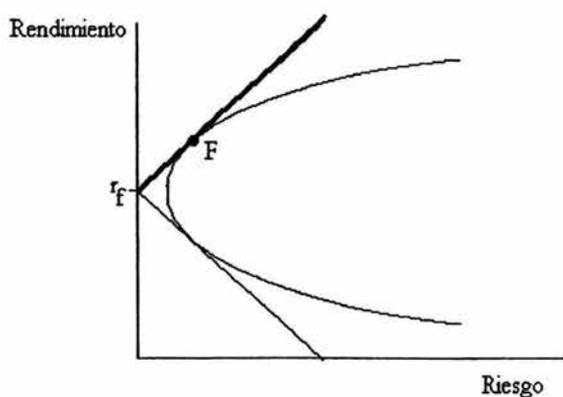


Figura 2.8

En esta frontera eficiente nueva, cualquier punto sobre ella puede ser expresado como una combinación del punto F y el activo libre de riesgo; en consecuencia, mediante el intercambio de los pesos entre estos dos se puede generar completamente la frontera eficiente. La cartera de inversión representada por el punto tangente puede ser pensada como una cartera hecha de varios activos y vendida como unidad. Este resultado es remarcado mediante la siguiente proposición.

Teorema de un fondo de inversión. Hay un solo fondo F de activos con riesgo tal que todo fondo eficiente puede ser construido como una combinación del fondo F y del activo libre de riesgo.

En la figura 2.9 se pueden observar, en una sola gráfica, las dos formas de la frontera eficiente, el punto T de varianza mínima, la frontera eficiente cuando no se ha incluido el activo libre de riesgo y S punto de tangencia entre las dos formas de la frontera eficiente. En el caso de la frontera eficiente con la inclusión de un activo libre de riesgo, el segmento de recta entre r_f y S corresponde a las combinaciones de los dos activos en las que $w_{rf} > 0$; mientras que en el segmento que inicia en S y continua ilimitadamente corresponde a las combinaciones de los mismos dos activos pero con $w_{rf} < 0$.

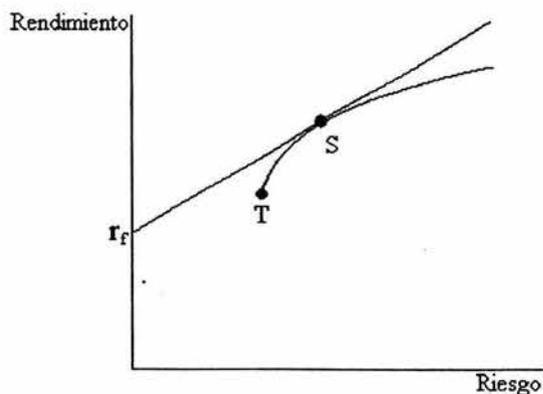


Figura 2.9

3. ALGORITMOS GENÉTICOS

3.1 Introducción a los algoritmos genéticos

La *computación evolutiva* se originó a partir de la intención de resolver problemas de optimización con base en los conocimientos sobre evolución. Algunos científicos comenzaron a estudiar los sistemas evolutivos durante las décadas de 1950 y 1960, con el propósito de poder usarlos como herramienta para solucionar problemas de optimización. La idea general detrás de éstos sistemas, es evolucionar una población de candidatos a ser solución de un problema dado mediante operadores.

Dentro de la computación evolutiva, tres de las técnicas más conocidas son: las redes neuronales, la lógica borrosa y los algoritmos genéticos. Las redes neuronales, basadas en la estructura y modo de operación del cerebro humano, son modelos computacionales no lineales que buscan la forma de reproducir características humanas como el aprendizaje, la asociación y la abstracción. La lógica borrosa, por su parte, tiene como objetivo modelar de forma aproximada el raciocinio humano, pretendiendo desarrollar sistemas computacionales capaces de tomar decisiones racionales en un ambiente de incertidumbre y precisión.

Otra de las técnicas más reconocidas son los algoritmos genéticos. El primer registro que se tiene de la utilización del término, es en una publicación de Bagley. En su texto, dicho autor diseñó algoritmos genéticos para buscar conjuntos de parámetros en funciones de evaluación de juegos. Sin embargo, el científico que es reconocido como el creador de los algoritmos genéticos es John Holland, quien los desarrolló durante las décadas de 1960 y 1970. De forma curiosa, el propósito original de Holland no era diseñar algoritmos para resolver problemas concretos, como lo pretendía la computación evolutiva;

él pretendía estudiar formalmente el fenómeno de la adaptación natural y, en la medida que fuera posible, aplicar esos mecanismos a los sistemas computacionales.

En su libro *Adaptación de Sistemas Naturales y Artificiales* publicado en 1975, Holland presentó un algoritmo genético creado a partir de la evolución biológica. Dicho algoritmo genético era una técnica para trasladarse de una población de cromosomas a una nueva población, por medio de la utilización de un sistema similar a la selección natural y operadores de cruce, mutación e inversión inspirados en la genética. En ese algoritmo genético, cada cromosoma era constituido por genes o bits; el operador de selección elegía a los cromosomas de la población con mayor capacidad. A continuación, el operador cruce extraía partes de algunos cromosomas para la creación de nuevos individuos, simulando así la combinación biológica. La mutación se encargaba de modificar el valor de un gen en determinadas posiciones del cromosoma, pero siempre de forma aleatoria. Finalmente, la inversión, como su nombre lo indica, invertía el orden de una sección del cromosoma.

También John Holland fue el primero en consolidar una base teórica firme para justificar los procedimientos de la computación evolutiva. Esta teoría se fundamenta en la noción de los “esquemas”, estructura sobre la que se han elaborado la mayoría de los trabajos teóricos sobre algoritmos genéticos.

Así, una primera definición sobre los algoritmos genéticos los describiría como algoritmos matemáticos inspirados en los procesos de la evolución natural y la recombinación genética. La técnica de los algoritmos genéticos ofrece un mecanismo de búsqueda adaptativa basada en el principio de Darwin de reproducción y supervivencia de los más aptos.

Ahora sería adecuado hablar un poco, aunque de modo muy básico, sobre los aspectos biológicos que son la base de los algoritmos genéticos.

Por un lado, es necesario conocer sobre la genética, el estudio de la herencia. Los genetistas investigan cómo operan los genes y cómo son transmitidos de padres a hijos. Existen muchos campos dentro de la genética como son la citogenética, genética molecular, genética humana, genética de población, etc.

En los humanos y en otros seres vivos, las células contienen grupos importantes de material especial que contienen la información hereditaria. A estas estructuras se les llama cromosomas, los cuales están formados por genes. El proceso básico de codificar información dentro de genes y cromosomas es simple, pero los posibles resultados casi ilimitados.

La composición genética de cada organismo es llamada genotipo. El genotipo determina y limita muchos aspectos del desarrollo y la supervivencia. El genotipo de un organismo, es quien determina principalmente su forma de responder a las condiciones ambientales. Las propiedades del organismo que resultan de sus encuentros con el ambiente son referidas como fenotipo.

Los humanos poseen 46 cromosomas colocados en parejas; de los 46, 23 vienen de la madre y 23 del padre. El proceso por el cual el número de cromosomas se reduce de 46 a 23 es llamado meiosis. En este proceso, los miembros de cada pareja de cromosomas se juntan intercambiando material genético; de esta forma se reduce a la mitad el número de cromosomas. Posteriormente se da un cruce de información al unirse la información del padre y la información de la madre, completando una vez más el número cromosómico propio de la especie.

Los seres vivos son sumamente fiables para copiar material genético, pero no perfectos. Ocasionalmente, cuando se copia el material genético resultan pequeñas imperfecciones llamadas mutaciones. La tasa de mutación es muy baja, pero esta agrega

una **diversidad** genética adicional a las poblaciones. A corto plazo, el efecto de una **mutación** sobre una población dada es insignificante; pero a largo plazo, puede guiar a cambios **significativos** en la población.

El **cruzamiento** y la **mutación**, que pueden no ser heredados de los padres al hijo, son **variaciones** genéticas que les permiten a algunos organismos adaptarse al ambiente. Si un organismo aprende las conductas que le permiten adaptarse y, mejor aún, sobrevivir en su ambiente, es bueno, pero no es una **variación** genética; esto es el **fenotipo**. Por esto, el **genotipo** es el punto crítico en la evolución.

El campo en la genética al que están más unidos los algoritmos genéticos es el de **genética de población**. Investigadores en este campo estudian las relaciones de la evolución, la mezcla de **material genético** en las especies y métodos de adaptación ambiental.

Por otro lado, es necesario hablar de Darwin, la adaptación y el proceso de **selección natural**. Él comprendió que la mayoría de los organismos, aunque producen **descendencia** que vive hasta madurar, no provocan que el número de individuos de su especie varíe demasiado en un periodo corto de tiempo. Comprendió que la muerte juega un papel **importante** en el diseño de la naturaleza. Darwin concluyó que la naturaleza **excluye** a los **insuficientemente** adaptados y llamó a este proceso **selección natural**.

Los individuos con **variaciones favorables** son más probables para sobrevivir y producir **descendencia**. De esta forma, las características de los individuos más aptos son más probables para ser pasadas de una generación a la siguiente.

Cada uno de los aspectos biológicos que se han comentado, seguramente de modo muy escueto según el criterio de un biólogo, son suficientes para que la conexión tan estrecha que existe entre la genética y los algoritmos genéticos, sea percibida de una forma clara.

3.2 Estructura de los algoritmos genéticos

3.2.1 Procedimiento de los sistemas evolutivos

Como se mencionó antes, el término sistemas evolutivos es usado para referirse a las técnicas que se basan en los principios de evolución. La estructura algorítmica en pseudocódigo de los sistemas evolutivos es la siguiente:

Iniciar

Hacer $t = 0$

Inicializar $P(t)$

Evaluar $P(t)$

Mientras (un criterio de finalización no se cumpla) *hacer*

Iniciar

Hacer $t = t + 1$

Seleccionar $P(t)$ de $P(t - 1)$

Alterar $P(t)$

Evaluar $P(t)$

Fin

Fin

Un sistema evolutivo es la implementación de un algoritmo probabilístico con la capacidad de operar a partir de una población de individuos, $P(t) = \{x_1^t, x_2^t, \dots, x_n^t\}$, para cada iteración de t . Un individuo en la población, entendido como una estructura de datos S , representa una solución potencial del problema. Cada solución x_i^t es evaluada para otorgarle una medida de aptitud; con base en esa medida se crea una nueva población (la iteración $t + 1$) donde los más aptos serán más propensos a ser seleccionados (paso de

selección). Mediante operadores, la nueva población sufrirá modificaciones y algunos individuos cederán su lugar a individuos nuevos (paso de alteración). Las modificaciones más comúnmente utilizadas son la mutación y el cruzamiento. La mutación, $m_i : S \rightarrow S$, crea nuevos individuos por medio de la modificación de alguna característica en un individuo en particular; el cruzamiento, $c_j : S \times S \times \dots \times S \rightarrow S$, crea nuevos individuos mediante la combinación de segmentos de al menos dos individuos. Posteriormente a la ejecución de un número de iteraciones del algoritmo, se pretende que el sistema evolutivo converja a la población en la que todos los individuos son el mismo, quien deberá representar la solución óptima del problema.

3.2.2 Procedimiento de los algoritmos genéticos

El vocabulario utilizado en los algoritmos genéticos, es en su mayoría adquirido a partir de la genética. Por ejemplo, los individuos de una población son llamados genotipos o cromosomas, aunque también se les puede llamar cadenas o estructuras. Los cromosomas están conformados por unidades, genes o características, que son ubicadas en una sucesión lineal; cada gen controla la herencia de una de varias características.

Debido a que un cromosoma representa a una solución potencial del problema, la ejecución del proceso de evolución de una población de cromosomas corresponde a la búsqueda a través de un espacio de soluciones potenciales. En cada búsqueda se intenta conciliar dos objetivos, ellos son la explotación de las mejores soluciones y la exploración del espacio de búsqueda. Por esta razón, se dice que los algoritmos genéticos son una técnica de búsqueda en múltiples direcciones, ya que en ellos se mantiene una población de soluciones potenciales, que facilita la acumulación de información y permite el intercambio entre las direcciones exploradas. De ese modo, la población sufre una evolución simulada

que favorece la reproducción de las soluciones relativamente buenas y propicia la exclusión de las relativamente malas. Una función objetivo, simulando el papel que tiene el medio ambiente en la naturaleza, es la forma de discriminar entre soluciones buenas y malas; a esta función se le llama función de aptitud, de desempeño, de capacidad o de evaluación.

Para corroborar que la estructura de los algoritmos genéticos se apega a la estructura de un sistema evolutivo, se puede observar que en la iteración t , un algoritmo genético mantiene una población de cromosomas (vectores), $P(t) = (x'_1, x'_2, \dots, x'_n)$. Cada x'_i es evaluada para asignarle su medida de aptitud. A partir de dicha medida, una nueva población será formada, induciendo a la selección de los individuos más aptos. Por medio de los operadores genéticos, algunos miembros de esta población nueva serán alterados para formar soluciones nuevas. El cruzamiento, combina las características de dos o más cromosomas padres para formar dos o más hijos por medio del intercambio de los segmentos respectivos de los padres. Por ejemplo, si el operador cruzamiento es aplicado a los vectores (a_1, b_1, c_1, d_1) y (a_2, b_2, c_2, d_2) , entonces el cruce de cromosomas después del segundo gen produciría a los hijos (a_1, b_1, c_2, d_2) y (a_2, b_2, c_1, d_1) . Aplicando el operador cruzamiento se facilita el intercambio de información entre distintas soluciones potenciales. Con respecto a la mutación, este operador altera uno o más genes de un cromosoma seleccionado, a partir de un cambio aleatorio que tiene una probabilidad de ocurrencia igual a una tasa de mutación considerada. El propósito de la aplicación del operador mutación es provocar una variabilidad adicional dentro de la población.

Como resumen de lo presentado hasta ahora, se puede señalar que el punto de partida, e incluso el punto crítico, de los algoritmos genéticos es la representación genética para las soluciones potenciales del problema o, en otras palabras, la codificación de

soluciones. Posteriormente, los algoritmos genéticos ejecutan cinco pasos de forma repetitiva: forman aleatoriamente una población de soluciones posibles del problema; calculan la aptitud de cada individuo en la población; seleccionan individuos para convertirse en padres de la siguiente generación; crean una segunda generación de hijos provenientes de los posibles padres; y regresan al segundo paso para calcular la aptitud de los nuevos individuos que permitirá continuar con la evolución hasta que se cumpla el criterio de finalización.

3.2.3 Aplicación de los algoritmos genéticos en un problema sencillo

Como se expuso en la sección anterior, el punto de partida para usar la técnica de los algoritmos genéticos como una herramienta de solución es representar el problema de tal modo que el algoritmo genético pueda trabajar con él. Usualmente, se utiliza una codificación binaria para representar a las soluciones potenciales del problema por medio de una cadena de dígitos binarios.

Supóngase que el problema es encontrar un valor para x tal que $C_{31}^x = 2,629,575$, es decir, se desea conocer un valor de x tal que las combinaciones de 31 elementos en x sean exactamente 2,629,575. Este ejemplo sencillo, servirá para mostrar la forma en que un problema puede ser representado para resolverse mediante un algoritmo genético.

El sistema binario o base dos, es una alternativa para el sistema convencional decimal o base 10. Se usa base dos para simplificar, ya que sólo requiere dos dígitos, 0 y 1. Así como la secuencia de los dígitos es importante en la notación decimal, esta también es importante en la notación binaria. El número decimal 1563 está compuesto por 1×1000 más 5×100 más 6×10 más 3×1 . Las cuatro posiciones (1000, 100, 10, 1) corresponden a

10^3 , 10^2 , 10^1 y 10^0 . De igual forma, el número binario 100 corresponde a 1×2^2 , 0×2^1 y 0×2^0 ; éste se puede describir como 1×4 más 0×2 más 0×1 , es decir, es el 4 decimal.

Para el problema, las soluciones potenciales serán codificadas como cadenas de cinco dígitos binarios, es decir, son permitidos todos los números binarios entre 00000 y 11111. La decisión de considerar cinco posiciones, se debe a que todas las soluciones potenciales del problema son representadas en base dos con a lo más cinco posiciones.

En cuanto al paralelismo biológico, la cadena de bits puede ser pensada como la estructura del cromosoma. Los 0s y los 1s corresponden a los genes dentro del cromosoma. De esta forma, con la codificación de las soluciones potenciales resuelta, se siguen los cinco pasos de la estructura de los algoritmos genéticos.

Paso 1. Se crea una población aleatoria de posibles soluciones del problema. Supóngase que se usa una población de tamaño 4; para ello se debe tener un generador de números aleatorios para formar la población inicial de 4 cadenas de longitud 5. También se puede proceder a través de cada cadena, bit por bit, como si se lanzara una moneda para definir si el bit debe ser 0 ó 1. Supóngase que la población consiste de los siguientes cuatro individuos: 00011, 01111, 11001 y 11111. Decodificando las soluciones iniciales del problema, se sabe que los números representados son 3, 15, 25 y 31.

Paso 2. Se calcula el desempeño, o la aptitud, de cada individuo de la población. Para usar los algoritmos genéticos, se elige una de entre varias formas para asignar un valor

de aptitud a cada cadena. En este caso:
$$\text{Aptitud} = 400,000,000 - \left| \frac{31!}{x!(31-x)!} - 2,629,575 \right|$$

Generalmente, los algoritmos genéticos son implementados de tal forma que la aptitud es un valor para ser maximizado. Es por ello que la función de aptitud adquirió una composición que pudiera parecer extraña. Sin embargo, la función será maximizada cuando

la expresión afectada por el valor absoluto sea cero; es decir, cuando $C_{31}^x = 2,629,575$. Ya que los algoritmos genéticos maximizan, se ha creado una función positiva de aptitud por optimizar que implícitamente resuelva el problema original. Con una cadena cuyo valor sea el óptimo del problema original, la aptitud será de 400,000,000; con una cadena cuyo valor no sean óptimo del problema original, la aptitud siempre será menor que 400,000,000.

A partir de la función creada, se calculan los valores de aptitud para las cuatro cadenas que integran la población. Los valores se muestran en la tabla 3.1.

Cadena Binaria	x	Aptitud
00011	3	397,374,920
01111	15	102,089,380
11001	25	398,106,706
11111	31	397,370,426

Tabla 3.1

No obstante que la función de aptitud está realizando lo que se requiere para resolver el problema, puede haber más y mejores funciones de aptitud para este problema.

Paso 3. Se seleccionan individuos para convertirse en padres de la siguiente generación. Básicamente, a los individuos con mejor desempeño debe otorgárseles el privilegio de la paternidad. Los individuos con un pobre desempeño no obtienen este privilegio. Existe una variedad de esquemas para ejecutar la selección.

Supóngase que se decide, como criterio para seleccionar a los individuos, eliminar a la cadena de peor desempeño para otorgarle su lugar a una copia de la cadena con mejor desempeño. De acuerdo a los resultados mostrados en la tabla 3.1, la segunda cadena será reemplazada por una copia de la tercera. Como consecuencia, los posibles padres serían las siguientes cadenas: 00011, 11001, 11001 y 11111.

Paso 4. Se integra una nueva generación con hijos provenientes de los posibles padres. En este paso se aplican los distintos operadores genéticos implementados. El

cruzamiento es uno de los dos operadores más utilizados, está inspirado en su contraparte biológica. Para aplicarlo, aleatoriamente se eligen dos cadenas dentro de los posibles padres. Sean la primera y la cuarta cadena, 00011 y 11111, las elegidas. En seguida, se elige aleatoriamente un punto en el cual las dos cadenas deben ser cortadas. Imagínese que el punto elegido fue el segundo. Las dos cadenas ahora se verían como: 00-011 y 11-111, donde el guión separa la cabeza de la cola de la cadena. Tercero, un número aleatorio indicará si el cruzamiento debe ser realizado o no. Por ejemplo, se puede indicar en el algoritmo, mediante una tasa de ocurrencia, que el cruzamiento se realizará aproximadamente el 60% de las ocasiones. Si el cruzamiento en cuestión está en el 60% se debe realizar, por lo que se procede a intercambiar las dos colas de las dos cadenas, dejando la cabeza de la primera cadena con la cola de la segunda cadena y viceversa. Ahora las cadenas son 00111 y 11011, las cuales corresponden a los valores decimales 7 y 27.

La mutación es el segundo operador más utilizado, pero con una probabilidad de ocurrencia mucho menor a la de cruzamiento. Con la mutación, un bit particular en una cadena particular es aleatoriamente elegido. Supóngase que se eligió el quinto bit del segundo cromosoma, entonces como consecuencia la cadena ahora es 11000 ó 24 decimal.

Paso 5. Regresando al paso 2, se obtienen los resultados mostrados en la tabla 3.2; en ella se observa que se han encontrado las dos soluciones óptimas del problema, 7 y 24, ya que $C_{31}^7 = 2,629,575$ y $C_{31}^{24} = 2,629,575$.

Cadena Binaria	x	Aptitud
00111	7	400,000,000
11000	24	400,000,000
11001	25	398,106,706
11011	27	397,401,890

Tabla 3.2

Los pasos 2 a 4 se siguen realizando hasta que la población converja. Se dice que la población *convergió totalmente* cuando las cadenas son idénticas en todas las posiciones. Cuando esto sucede, el cruzamiento no afecta; la única forma de mejorar esta fase es con la mutación. Pero, si las cadenas son todas óptimas, la mutación sólo empeorará la situación.

3.3 Operatividad de los algoritmos genéticos

3.3.1 Los algoritmos genéticos operando

Como se dijo antes, los algoritmos genéticos se usan en problemas de maximización. Cuando el problema es minimizar la función f , se puede maximizar la función g , donde $g = -f$. También se suele asumir que la función objetivo es positiva, es decir, cualquier solución potencial evaluada en la función objetivo siempre otorgará un resultado positivo; para ello se le suma una constante positiva C suficientemente grande.

Entonces como punto de partida supóngase que se desea maximizar una función de k variables, $f(x_1, x_2, \dots, x_k): R^k \rightarrow R$; cada variable x_i tiene dominio $D_i = [a_i, b_i] \subseteq R$ y $f(x_1, x_2, \dots, x_k) > 0$ para todo $x \in D_1 \times D_2 \times \dots \times D_k$. Además, supóngase que en el problema en cuestión se ha establecido que cada variable x_i debe tener una precisión de h cifras decimales.

Con el fin de alcanzar tal precisión, cada dominio D_i es dividido en $(b_i - a_i) \cdot 10^h$ intervalos de igual tamaño y se busca al entero más pequeño, l_i , que verifique la desigualdad $(b_i - a_i) \cdot 10^h \leq 2^{l_i} - 1$. De esta forma, cada variable x_i codificada como una cadena binaria de longitud l_i , logra satisfacer la precisión solicitada. A partir de los supuestos realizados, la siguiente expresión permite decodificar las cadenas binarias:

$$x_i = a_i + VDC \cdot \frac{b_i - a_i}{2^{l_i} - 1} \quad (3.1)$$

donde VDC representa el valor decimal de la cadena binaria.

Con base en la forma de codificar a los cromosomas, cada uno es codificado como una cadena binaria de longitud $l = \sum_{i=1}^k l_i$, donde el mapa de bits l_1 es un valor en el intervalo $[\alpha_1, b_1]$, el mapa de bits l_2 en un valor del rango $[\alpha_2, b_2]$ y así sucesivamente.

Es momento de inicializar la población, para ello se crean n individuos aleatoriamente. Entonces $P = \{v_1, v_2, \dots, v_n\}$ es el conjunto de individuos generados.

El proceso de selección de una nueva población se fundamenta en el cálculo de una distribución de probabilidad basada en los valores de aptitud. La construcción de tal distribución se realiza ejecutando los siguientes cálculos:

1. Se calcula el valor de aptitud, $apt(v_i)$, para cada cromosoma v_i , $i = 1, \dots, n$.
2. Se encuentra la aptitud total de la población:

$$A = \sum_{i=1}^n apt(v_i) \quad (3.2)$$

3. Se calcula la probabilidad de selección p_i para cada v_i con $i = 1, \dots, n$:

$$p_i = \frac{apt(v_i)}{A} \quad (3.3)$$

4. Se calcula una probabilidad acumulativa q_i para cada v_i con $i = 1, \dots, n$:

$$q_i = \sum_{j=1}^i p_j \quad (3.4)$$

El proceso de selección se ejecuta con la ayuda de una serie $R_s = \{r_1, r_2, \dots, r_n\}$ de números aleatorios en el rango $[0,1]$. Para cada $r_m \in R_s$ con $m = 1, 2, \dots, n$, si $r_m < q_1$, entonces se selecciona a v_1 ; en caso contrario se selecciona a v_i con $2 \leq i \leq n$ y tal que $q_{i-1} < r_m < q_i$.

El siguiente paso es la aplicación del operador cruzamiento sobre la nueva población. El número aproximado de ejecuciones del operador, está ligado a un parámetro del algoritmo conocido como probabilidad de cruzamiento, p_c , y puede ser calculado como:

$$p_c \cdot n \quad (3.5)$$

Para aplicar el cruzamiento es necesario generar una nueva serie $R_C = \{r_1, r_2, \dots, r_n\}$ de números aleatorios en el rango $[0,1]$. Para cada $r_m \in R_C$ con $m = 1, 2, \dots, n$, si $r_m < p_c$, entonces se selecciona al cromosoma v_m para el cruzamiento.

Se procede a formar parejas (v_i, v_j) de cromosomas seleccionados. Para cada pareja, se genera un número entero aleatorio, u , en el rango $[1, l-1]$. El número u indica la posición de cruce. Por ejemplo, si $(b_1 b_2 \dots b_u b_{u+1} \dots b_l)$ y $(c_1 c_2 \dots c_u c_{u+1} \dots c_l)$ fueron seleccionados, serán remplazados por sus hijos $(b_1 b_2 \dots b_u c_{u+1} \dots c_l)$ y $(c_1 c_2 \dots c_u b_{u+1} \dots b_l)$.

Para la aplicación del operador mutación, se realiza una inspección bit a bit. Ahora, el número esperado de bits por mutar depende de la probabilidad de mutación p_m :

$$p_m \cdot n \cdot l \quad (3.6)$$

Para aplicar la mutación es necesario generar una serie $R_M = \{r_1, r_2, \dots, r_{n \cdot l}\}$ de números aleatorios en el rango $[0,1]$. Para cada $r_m \in R_M$ con $m = 1, 2, \dots, n \cdot l$, si $r_m < p_m$, entonces el bit m mutará; es decir, si el bit m tenía asignado el valor 0, entonces su valor cambiará a 1 y viceversa.

Una vez que se ha terminado de ejecutar el operador mutación, una nueva población está lista para ser evaluada. Esta evaluación, será utilizada para construir una nueva distribución de probabilidad que marcará el inicio de una nueva iteración del algoritmo genético.

3.3.2 Corrida de un algoritmo genético

Esta sección está destinada a la presentación de una corrida de un algoritmo genético, tomada de Michalewicz [9]. La razón principal por la que se muestra esta ejecución en particular, se debe a que representa un magnífico ejemplo didáctico. Su autor asume que $f(x_1, x_2) = 21.5 + x_1 \cdot \sin(4\pi x_1) + x_2 \cdot \sin(20\pi x_2)$ es la función a optimizar con $-3.0 \leq x_1 \leq 12.1$ y $4.1 \leq x_2 \leq 5.8$; además considera $n = 20$, $p_c = 0.25$, $p_m = 0.01$ y $h = 4$.

Considerando el dominio de la variable x_1 de longitud 15.1 y la precisión requerida, entonces el intervalo $D_1 = [-3.0, 12.1]$ se divide en $15.1 \cdot 10000$ intervalos iguales. Debido a que $2^{17} < 151000 \leq 2^{18} - 1$, el cromosoma en su primera parte constará de 18 bits. Considerando el dominio de la variable x_2 , de longitud 1.7 y considerando nuevamente la precisión requerida, el intervalo $D_2 = [4.1, 5.8]$ se divide en $1.7 \cdot 10000$ intervalos iguales. Como consecuencia, se necesitan 15 bits para la segunda parte del cromosoma porque $2^{14} < 17000 \leq 2^{15} - 1$. Entonces, cada v_i tendrá una longitud de 33 bits.

De esta forma, $v = 001000100000110101|111011011111011$ corresponde a $(x_1, x_2) = (-0.991471, 5.680258)$. Con base en la ecuación (3.1), se calcula a partir de los 18 bits iniciales el valor de x_1 y a partir de los 15 bits finales x_2 , de donde se obtiene:

$$\begin{aligned}x_1 &= -3.0 + VDC(001000100000110101) \cdot \frac{12.1 - (-3.0)}{2^{18} - 1} \\ &= -3.0 + 34869 \cdot \frac{15.1}{262143} = -3.0 + 2.008529 = -0.991471\end{aligned}$$

$$\begin{aligned}x_2 &= 4.1 + VDC(111011011111011) \cdot \frac{5.8 - 4.1}{2^{15} - 1} \\ &= 4.1 + 30459 \cdot \frac{1.7}{32767} = 4.1 + 1.580258 = 5.680258\end{aligned}$$

El valor de aptitud de v es: $f(-0.991471, 5.680258) = 16.020855$.

La población inicial creada de forma aleatoria es:

$v_1=100110100000001111111010011011111$	$v_{11}=01100111111011010110000110111000$
$v_2=111000100100110111001010100011010$	$v_{12}=11010001011110110100010101000000$
$v_3=000010000011001000001010111011101$	$v_{13}=111011111010001000110000001000110$
$v_4=100011000101101001111000001110010$	$v_{14}=010010011000001010100111100101001$
$v_5=000111011001010011010111111000101$	$v_{15}=111011101101110000100011111011110$
$v_6=000101000010010101001010111111011$	$v_{16}=110011110000011111100001101001011$
$v_7=001000100000110101111011011111011$	$v_{17}=011010111111001111010001101111101$
$v_8=100001100001110100010110101100111$	$v_{18}=011101000000001110100111110101101$
$v_9=010000000101100010110000001111100$	$v_{19}=000101010011111111110000110001100$
$v_{10}=000001111000110000011010000111011$	$v_{20}=101110010110011110011000101111110$

Tabla 3.3

En el paso de evaluación se obtienen los siguientes resultados:

$\text{apt}(v_1) = f(6.084492, 5.652242) = 26.019465$	$\text{apt}(v_{11}) = f(3.130078, 4.996097) = 23.410744$
$\text{apt}(v_2) = f(10.348434, 4.380264) = 7.580032$	$\text{apt}(v_{12}) = f(9.356179, 4.239457) = 15.011719$
$\text{apt}(v_3) = f(-2.516603, 4.390381) = 19.526420$	$\text{apt}(v_{13}) = f(11.134646, 5.378671) = 27.316719$
$\text{apt}(v_4) = f(5.278638, 5.593460) = 17.406760$	$\text{apt}(v_{14}) = f(1.335944, 5.151378) = 19.876290$
$\text{apt}(v_5) = f(-1.255173, 4.734458) = 25.341251$	$\text{apt}(v_{15}) = f(11.089025, 5.054515) = 30.060311$
$\text{apt}(v_6) = f(-1.811725, 4.391937) = 18.100404$	$\text{apt}(v_{16}) = f(9.211598, 4.993762) = 23.867188$
$\text{apt}(v_7) = f(-0.991471, 5.680258) = 16.020855$	$\text{apt}(v_{17}) = f(3.367514, 4.571343) = 13.696179$
$\text{apt}(v_8) = f(4.910618, 4.703018) = 17.959593$	$\text{apt}(v_{18}) = f(3.843020, 5.158226) = 15.414224$
$\text{apt}(v_9) = f(0.795406, 5.381472) = 16.127756$	$\text{apt}(v_{19}) = f(-1.746635, 5.395584) = 20.095883$
$\text{apt}(v_{10}) = f(-2.554851, 4.793707) = 21.278378$	$\text{apt}(v_{20}) = f(7.935998, 4.757338) = 13.666961$

Tabla 3.4

De la tabla 3.4, se observa que v_{15} es el individuo más fuerte y v_2 es el más débil.

La aptitud total de la población es, de acuerdo a la ecuación 3.2, $A = 387.777132$.

A partir del valor A , se construye la función de distribución de probabilidad. Para ello, se calcula la probabilidad de selección p_i para cada cromosoma de acuerdo a la fórmula 3.3.

$p_1 = 0.067099$	$p_5 = 0.065350$	$p_9 = 0.041590$	$p_{13} = 0.070444$	$p_{17} = 0.035320$
$p_2 = 0.019547$	$p_6 = 0.046677$	$p_{10} = 0.054873$	$p_{14} = 0.051257$	$p_{18} = 0.039750$
$p_3 = 0.050355$	$p_7 = 0.041315$	$p_{11} = 0.060372$	$p_{15} = 0.077520$	$p_{19} = 0.051823$
$p_4 = 0.044889$	$p_8 = 0.046314$	$p_{12} = 0.038712$	$p_{16} = 0.061549$	$p_{20} = 0.035244$

Tabla 3.5

La tabla 3.5 muestra las probabilidades de selección calculadas para cada uno de los cromosoma que conforman la población.

Mediante la fórmula 3.4 se calculan las probabilidades acumulativas, q_i , para cada cromosoma; dichas probabilidades son mostradas en la tabla 3.6.

$q_1 = 0.067099$	$q_5 = 0.247240$	$q_9 = 0.423136$	$q_{13} = 0.647537$	$q_{17} = 0.873182$
$q_2 = 0.086646$	$q_6 = 0.293917$	$q_{10} = 0.478009$	$q_{14} = 0.698794$	$q_{18} = 0.912932$
$q_3 = 0.137001$	$q_7 = 0.335232$	$q_{11} = 0.538381$	$q_{15} = 0.776314$	$q_{19} = 0.964756$
$q_4 = 0.181890$	$q_8 = 0.381546$	$q_{12} = 0.577093$	$q_{16} = 0.837862$	$q_{20} = 1.000000$

Tabla 3.6

Ahora se crea el conjunto R_s que está integrado por $n = 20$ elementos. La secuencia de los 20 números aleatorios en el rango $[0,1]$ son mostrados en la tabla 3.7 (deben ser leídos de arriba hacia abajo y de izquierda a derecha):

0.513870	0.947628	0.494773	0.277226	0.465682
0.175741	0.171736	0.124720	0.368071	0.646473
0.308652	0.702231	0.703899	0.983437	0.767139
0.534534	0.226431	0.389647	0.005398	0.780237

Tabla 3.7

En el caso de $r_1 = 0.513870$, dado que es mayor que q_{10} y menor que q_{11} , se debe seleccionar el cromosoma v_{11} para formar parte de la nueva población; en el caso de $r_2 = 0.175741$, debido a que es más grande que q_3 y menor que q_4 , se concluye que el cromosoma v_4 es seleccionado para formar parte de la nueva población. Así se procede con los restantes 18 números aleatorios; la nueva población creada se muestra en la tabla 3.8.

$v'_1 = 011001111110110101100001101111000 = v_{11}$
$v'_2 = 100011000101101001111000001110010 = v_4$
$v'_3 = 00100010000011010111101101111011 = v_7$
$v'_4 = 011001111110110101100001101111000 = v_{11}$
$v'_5 = 00010101001111111110000110001100 = v_{19}$
$v'_6 = 100011000101101001111000001110010 = v_4$
$v'_7 = 111011101101110000100011111011110 = v_{15}$
$v'_8 = 000111011001010011010111111000101 = v_5$
$v'_9 = 011001111110110101100001101111000 = v_{11}$
$v'_{10} = 000010000011001000001010111011101 = v_3$

$v'_{11} = 111011101101110000100011111011110 = v_{15}$
$v'_{12} = 010000000101100010110000001111100 = v_9$
$v'_{13} = 000101000010010101001010111111011 = v_6$
$v'_{14} = 100001100001110100010110101100111 = v_8$
$v'_{15} = 101110010110011110011000101111110 = v_{20}$
$v'_{16} = 100110100000001111111010011011111 = v_1$
$v'_{17} = 00000111100011000011010000111011 = v_{10}$
$v'_{18} = 111011111010001000110000001000110 = v_{13}$
$v'_{19} = 111011101101110000100011111011110 = v_{15}$
$v'_{20} = 110011110000011111100001101001011 = v_{16}$

Tabla 3.8

Se sabe que $p_c = 0.25$, entonces en promedio 5 de cada 20 cromosomas sufrirán el cruzamiento. Ahora se generan 20 números aleatorios¹⁴ en el intervalo $[0,1]$ (los elementos del conjunto R_C); si $r_i < 0.25$, se selecciona al cromosoma para el cruzamiento.

Los elementos del conjunto R_C generados se muestra en la tabla 3.9.

0.822951	0.346901	0.606758	0.166225	0.389248
0.151932	0.917204	0.785402	0.674520	0.200232
0.625477	0.519760	0.031523	0.758400	0.355635
0.314685	0.401154	0.869921	0.581893	0.826927

Tabla 3.9

Después de verificar las desigualdades indicadas, los cromosomas a los que se aplicará cruzamiento son: v'_2 , v'_{11} , v'_{13} y v'_{18} . Dado que el número de individuos elegidos es par, no existe inconveniente para formar las parejas. En caso contrario, se hubiera procedido a agregar uno o eliminar uno de los elegidos, pero esta elección también debe ser hecha aleatoriamente. Las parejas son formadas por cromosomas consecutivos; para cada una de las parejas, se genera un número entero aleatorio u en el intervalo $[1, 32]$. El número u indica la posición del punto de cruce.

Para la pareja v'_2 , v'_{11} , el número generado fue $u = 9$. Para la pareja v'_{13} , v'_{18} , se obtuvo que $u = 20$. En la tabla 3.10 se muestran, los resultados obtenidos.

Primera pareja de padres seleccionada	$v'_2 = 100011000 101101001111000001110010$ $v'_{11} = 111011101 101110000100011111011110$
Hijos que reemplazarán a la primera pareja de padres	$v''_2 = 100011000 101110000100011111011110$ $v''_{11} = 111011101 101101001111000001110010$
Segunda pareja de padres seleccionada	$v'_{13} = 00010100001001010100 1010111111011$ $v'_{18} = 1110111110100010001 0000001000110$
Hijos que reemplazarán a la segunda pareja de padres	$v''_{13} = 00010100001001010100 0000001000110$ $v''_{18} = 1110111110100010001 1010111111011$

Tabla 3.10

La población, después de aplicar el cruzamiento, se muestra en la tabla 3.11.

$v'_1 = 011001111110110101100001101111000$	$v''_{11} = 111011101101101001111000001110010$
$v''_2 = 100011000101110000100011111011110$	$v'_{12} = 010000000101100010110000001111100$
$v'_3 = 00100010000011010111101101111011$	$v''_{13} = 00010100001001010100000001000110$
$v'_4 = 011001111110110101100001101111000$	$v'_{14} = 100001100001110100010110101100111$
$v'_5 = 00010101001111111110000110001100$	$v'_{15} = 101110010110011110011000101111110$
$v'_6 = 100011000101101001111000001110010$	$v'_{16} = 100110100000001111111010011011111$
$v'_7 = 111011101101110000100011111011110$	$v'_{17} = 000001111000110000011010000111011$
$v'_8 = 000111011001010011010111111000101$	$v''_{18} = 111011111010001000111010111111011$
$v'_9 = 011001111110110101100001101111000$	$v'_{19} = 111011101101110000100011111011110$
$v'_{10} = 000010000011001000001010111011101$	$v'_{20} = 110011110000011111100001101001011$

Tabla 3.11

Se supuso que la probabilidad de mutación es $p_m = 0.01$, entonces en promedio 1% de los bits podrá mutar. La mutación se aplica bit por bit en cada cromosoma; ya que existen $l \times n = 33 \times 20 = 660$ bits, se esperan aproximadamente 6.6 mutaciones por generación. Se genera la serie R_M de números aleatorios en el rango $[0,1]$; si $r_i < 0.01$, el bit sufre mutación. En este caso, sólo 5 de éstos números son menores que p_m .

Posición del bit	Número aleatorio	Posición del bit	Número aleatorio
112	0.000213	429	0.005425
349	0.009945	603	0.002836
418	0.008809		

Tabla 3.12

En la tabla 3.12 se muestran los resultados de las comparaciones realizadas, es decir, los bits que sufrirán mutación y su número aleatorio. La correspondencia de la posición del bit con el número de cromosoma y número de bit se muestran en la tabla 3.13.

Posición del bit	Número de cromosoma	Número de bit dentro del cromosoma
112	4	13
349	11	19
418	13	22
429	13	33
603	19	9

Tabla 3.13

La población al finalizar esta primera iteración se observa en la tabla 3.14. En ella se distinguen los bits, escritos en negritas, de los cromosomas que mutaron.

$v_1 = 01100111111011010110000110111000$	$v_{11} = 11101110110110101001011000001110010$
$v_2 = 100011000101110000100011111011110$	$v_{12} = 010000000101100010110000001111100$
$v_3 = 00100010000011010111101101111011$	$v_{13} = 000101000010010101000100001000111$
$v_4 = 01100111111001010110000110111000$	$v_{14} = 100001100001110100010110101100111$
$v_5 = 00010101001111111110000110001100$	$v_{15} = 101110010110011110011000101111110$
$v_6 = 100011000101101001111000001110010$	$v_{16} = 100110100000001111111010011011111$
$v_7 = 111011101101110000100011111011110$	$v_{17} = 000001111000110000011010000111011$
$v_8 = 000111011001010011010111111000101$	$v_{18} = 11101111101000100011101011111011$
$v_9 = 01100111111011010110000110111000$	$v_{19} = 111011100101110000100011111011110$
$v_{10} = 000010000011001000001010111011101$	$v_{20} = 110011110000011111100001101001011$

Tabla 3.14

Terminó una iteración. La evaluación de la nueva población inicia otra iteración.

$\text{apt}(v_1) = f(3.130078, 4.996097) = 23.410744$	$\text{apt}(v_{11}) = f(11.088621, 4.743434) = 33.351848$
$\text{apt}(v_2) = f(5.279042, 5.054515) = 18.201145$	$\text{apt}(v_{12}) = f(0.795406, 5.381472) = 16.127756$
$\text{apt}(v_3) = f(-0.991471, 5.680258) = 16.020855$	$\text{apt}(v_{13}) = f(-1.811725, 4.209937) = 22.692491$
$\text{apt}(v_4) = f(3.128235, 4.996097) = 23.412688$	$\text{apt}(v_{14}) = f(4.910618, 4.703018) = 17.959593$
$\text{apt}(v_5) = f(-1.746635, 5.395584) = 20.095883$	$\text{apt}(v_{15}) = f(7.935998, 4.757338) = 13.666961$
$\text{apt}(v_6) = f(5.278638, 5.593460) = 17.406760$	$\text{apt}(v_{16}) = f(6.084492, 5.652242) = 26.019465$
$\text{apt}(v_7) = f(11.089025, 5.054515) = 30.060311$	$\text{apt}(v_{17}) = f(-2.554851, 4.793707) = 21.278403$
$\text{apt}(v_8) = f(-1.255173, 4.734458) = 25.341251$	$\text{apt}(v_{18}) = f(11.134646, 5.666976) = 27.591058$
$\text{apt}(v_9) = f(3.130078, 4.996097) = 23.410744$	$\text{apt}(v_{19}) = f(11.059532, 5.054515) = 27.608488$
$\text{apt}(v_{10}) = f(-2.516603, 4.390381) = 19.526420$	$\text{apt}(v_{20}) = f(9.211598, 4.993762) = 23.867188$

Tabla 3.15

En la tabla 3.15 se muestra la aptitud de cada individuo. Debe hacerse notar que la aptitud de la nueva población es $A = 447.050052$, superior a la aptitud de la generación anterior, 387.777132. Además, el mejor cromosoma (v_{11}) tiene una mejor evaluación, 33.351848, que el mejor cromosoma (v_{15}) de la población previa, el cual obtuvo 30.060311. La población de la iteración 1000 y su aptitud son mostradas en las tablas 3.16 y 3.17.

$v_1=111011110110011011100101010111011$	$v_{11}=110101100000010010001100010110000$
$v_2=111001100110000100010101010111000$	$v_{12}=110101100010010010001100010110001$
$v_3=111011110111011011100101010111011$	$v_{13}=111011110111011011100101010111011$
$v_4=111001100010000110000101010111001$	$v_{14}=111001100110000100000101010111011$
$v_5=111011110111011011100101010111011$	$v_{15}=111001101010111001010100110110001$
$v_6=111001100110000100000100010100001$	$v_{16}=111001100110000101000100010100001$
$v_7=110101100010010010001100010110000$	$v_{17}=111001100110000100000101010111011$
$v_8=111101100010001010001101010010001$	$v_{18}=111001100110000100000101010111001$
$v_9=111001100010010010001100010110001$	$v_{19}=111101100010001010001110000010001$
$v_{10}=111011110111011011100101010111011$	$v_{20}=111001100110000100000101010111001$

Tabla 3.16

$\text{apt}(v_1)=f(11.120994, 5.092544)=30.307492$	$\text{apt}(v_{11})=f(9.623741, 4.427891)=35.477578$
$\text{apt}(v_2)=f(10.588808, 4.667376)=28.870407$	$\text{apt}(v_{12})=f(9.631114, 4.427943)=35.455157$
$\text{apt}(v_3)=f(11.124680, 5.092544)=30.325177$	$\text{apt}(v_{13})=f(11.124680, 5.092544)=30.325177$
$\text{apt}(v_4)=f(10.574177, 4.242415)=31.936129$	$\text{apt}(v_{14})=f(10.588808, 4.242518)=32.934221$
$\text{apt}(v_5)=f(11.124680, 5.092544)=30.325177$	$\text{apt}(v_{15})=f(10.606607, 4.653731)=30.743561$
$\text{apt}(v_6)=f(10.588808, 4.214606)=34.359757$	$\text{apt}(v_{16})=f(10.588865, 4.214606)=34.363137$
$\text{apt}(v_7)=f(9.631114, 4.427891)=35.457743$	$\text{apt}(v_{17})=f(10.588808, 4.242518)=32.934221$
$\text{apt}(v_8)=f(11.518161, 4.452846)=23.313872$	$\text{apt}(v_{18})=f(10.588808, 4.242415)=32.958611$
$\text{apt}(v_9)=f(10.574868, 4.427943)=34.397381$	$\text{apt}(v_{19})=f(11.518161, 4.472768)=19.677035$
$\text{apt}(v_{10})=f(11.124680, 5.092544)=30.325177$	$\text{apt}(v_{20})=f(10.588808, 4.242415)=32.958611$

Tabla 3.17

Por último Michalewicz [9] señala “Sin embargo, si observamos cuidadosamente el progreso durante la corrida, podemos descubrir que en las primeras generaciones el valor de aptitud de algunos cromosomas fue mayor que el valor 35.477578 del mejor cromosoma después de 1000 generaciones. Por ejemplo, el mejor cromosoma de la generación 396, tuvo aptitud de 38.827553”. Estos errores, efecto de la aleatoriedad, pueden ser evitados

mediante técnicas elitistas, al aplicar los operadores o al finalizar la iteración, que permiten la supervivencia del más apto en cada generación.

3.3.3 Fundamentos de la operatividad de los algoritmos genéticos

El funcionamiento de los algoritmos genéticos, está basado en el “teorema del esquema” creado por Holland y publicado en 1975. En su teorema, Holland postula que los algoritmos genéticos encuentran estructuras que representan a buenas soluciones, propician la permanencia de dichas estructuras y buscan mejorarlas al combinarlas unas con otras. Para mostrar la validez de esta idea, Holland analiza los fenómenos provocados por el proceso de selección, la aplicación del operador cruzamiento y la aplicación del operador mutación sobre las estructuras.

Formalmente, a las estructuras mencionadas en el párrafo anterior se les llama esquemas. Un esquema es entendido como una cadena de bits de una determinada longitud, en la que a cada bit o posición se le asigna una de las siguientes tres opciones: 0, 1 ó *. Cuando la posición de un esquema tiene asignado el valor 0 ó 1, se dice que el bit tiene un valor fijo; cuando la posición de un esquema tiene asignado el símbolo *, se dice que el bit no tiene un valor definido, ya que la posición ocupada por ese símbolo puede adquirir el valor 0 ó 1. Por ejemplo, considerando esquemas de longitud 7, el esquema (*011010) representa dos cadenas {(0011010), (1011010)}; el esquema (*0*1010) representa a cuatro cadenas {(0001010), (0011010), (1001010), (1011010)}; el esquema (1011010) representa sólo a la cadena (1011010); y el esquema (******) representa todas las cadenas de longitud 7.

Los esquemas poseen dos propiedades importantes: el orden y la longitud definida. El orden de un esquema E , $\alpha(E)$, es el número de bits con un valor fijo, es decir,

el número de posiciones con valor 0 ó 1. La longitud definida de un esquema E , $ld(E)$, es la distancia entre la primera y la última posición fija de la cadena; esta medida indica el menor tamaño posible al que se podría reducir la información de un esquema.

Por ejemplo, a partir de los esquemas $E_1 = (001*1**)$, $E_2 = (*1110*0)$ y $E_3 = (1*10001)$ se tiene que $\alpha(E_1) = 4$, $\alpha(E_2) = 5$ y $\alpha(E_3) = 6$. En cuanto a la longitud definida se tiene que $ld(E_1) = 5 - 1 = 4$, $ld(E_2) = 7 - 2 = 5$ y $ld(E_3) = 7 - 1 = 6$.

Algunos otros cálculos relacionados con los esquemas, que son dignos de tomarse en cuenta, son los relacionados con la cardinalidad de ciertos conjuntos creados a partir de los esquemas. Por ejemplo, cuando z denota al número de símbolos $*$ en el esquema, entonces 2^z cadenas son representadas por el esquema. También cuando se consideran cadenas de longitud l , se sabe que una cadena está representada por 2^l esquemas; mientras que el número total de esquemas existentes es 3^l .

Un cálculo que merece ser mencionado por separado, indica que en una población de tamaño n , se pueden representar entre 2^l y $n \cdot 2^l$ esquemas diferentes. Es importante resaltar esta afirmación debido a que es posible darle una interpretación más contundente; lo que indica es que un algoritmo genético a partir de la evaluación explícita de n individuos, realiza una evaluación implícita de entre 2^l y $n \cdot 2^l$ esquemas.

Otra característica de un esquema E , es el número de cadenas de la población que representa en el tiempo t , $I(E, t)$. A partir de esa característica, es posible realizar el cálculo de la aptitud del esquema E en el tiempo t , $apt(E, t)$. Esta aptitud se define como el promedio de las aptitudes de las cadenas en la población representadas por el esquema E . Asumiendo que hay p cadenas $\{v_1, v_2, \dots, v_p\}$ en la población representadas por el esquema E al tiempo t , la aptitud del esquema E en el tiempo t se calcula mediante la expresión:

$$apt(E, t) = \frac{\sum_{j=1}^p apt(v_j)}{p} \quad (3.7)$$

De acuerdo a lo comentado al inicio de ésta sección, se considera que el primer fenómeno sufrido por un esquema se da en el paso de selección. Suponiendo que $I(E, t)$ ha sido calculado, después del paso de selección se espera tener $I(E, t+1)$ cadenas representadas por el esquema E . Debe hacerse notar que para una cadena promedio representada por un esquema E , la probabilidad de su selección es igual al cociente de la aptitud del esquema entre la aptitud total de la población. Además el número de selecciones de cadenas simple es n , por lo que se tiene la siguiente igualdad:

$$I(E, t+1) = I(E, t) \cdot n \cdot \frac{apt(E, t)}{A(t)} \quad (3.8)$$

La fórmula 3.8 se puede reescribir tomando en cuenta que el promedio de la aptitud de la población se calcula como:

$$\overline{A(t)} = \frac{A(t)}{n} \quad (3.9)$$

entonces:

$$I(E, t+1) = I(E, t) \cdot \frac{apt(E, t)}{\overline{A(t)}} \quad (3.10)$$

La ecuación 3.10 es conocida como la ecuación de crecimiento reproductivo del esquema. Lo que indica esta ecuación es que el número de cadenas en la población representadas por el esquema, crece como la razón de la aptitud del esquema entre el promedio de aptitud de la población. Esto significa que para un esquema con aptitud mayor que el promedio de aptitud de la generación, su número de cadenas representadas en la siguiente generación aumentará; para un esquema con promedio de aptitud menor que el

promedio de aptitud de la generación, su número de cadenas representadas disminuirá; y para un esquema con promedio de aptitud idéntico al promedio de aptitud de la generación, el número de cadenas representadas no se alterará.

Para estimar el efecto de la ecuación de crecimiento a largo plazo, se asume que un esquema E se mantendrá con un promedio de aptitud superior en un $\varepsilon\%$ al promedio de aptitud de la generación; es decir, $apt(E,t) = \overline{A(t)} + \varepsilon \cdot \overline{A(t)}$. A partir de ese supuesto, se puede expresar el número de cadenas representadas por el esquema en el tiempo t , en función del número de cadenas representadas inicialmente por el esquema y de ε , como se muestra en la siguiente ecuación:

$$I(E,t) = I(E,0)(1 + \varepsilon)^t \quad (3.11)$$

donde $\varepsilon = \frac{apt(E,t) - \overline{A(t)}}{\overline{A(t)}}$. A partir de la expresión que permite calcular ε , se sabe que para un esquema con promedio de aptitud menor al promedio de aptitud de la generación, $\varepsilon < 0$; por el contrario, para un esquema con promedio de aptitud mayor al promedio de aptitud de la generación, $\varepsilon > 0$. Ahora, además de afirmar que un esquema con promedio de aptitud menor al promedio de aptitud de la generación, representa un número menor de cadenas en la siguiente generación, también se sabe que tal esquema representa un número de cadenas disminuido de forma exponencial en las siguientes generaciones.

Otros fenómenos que pueden afectar la supervivencia de un esquema a lo largo de la ejecución de un algoritmo, se presentan con la aplicación de los operadores genéticos: el cruzamiento y la mutación.

La probabilidad de que un esquema puede sobrevivir o ser destruido por medio de la aplicación del operador de cruzamiento, depende de su longitud definida. Cuando el punto de cruce seleccionado no se encuentra en medio del segmento que corresponde a la

longitud definida, el esquema sobrevive al cruzamiento. Dado que el lugar de cruzamiento es seleccionado uniformemente entre $l - 1$ lugares posibles, entonces la probabilidad de destrucción del esquema E es:

$$p_d(E) = \frac{ld(E)}{l-1} \quad (3.12)$$

y en consecuencia, la probabilidad de supervivencia del esquema es:

$$p_s(E) = 1 - \frac{ld(E)}{l-1} \quad (3.13)$$

También debe considerarse que solamente algunos cromosomas se ven afectados por el operador cruzamiento, cuya probabilidad de ocurrencia es p_c . Esto significa que la probabilidad de supervivencia de un esquema es:

$$p_s(E) = 1 - p_c \cdot \frac{ld(E)}{l-1} \quad (3.14)$$

Sin embargo, aún cuando el punto de cruce seleccionado esté en medio del segmento correspondiente a la longitud definida del esquema, el esquema tiene una probabilidad positiva de sobrevivir. Para que esto ocurra es necesario que los dos esquemas por cruzar coincidan en todas las posiciones posteriores al punto de cruce. Aunque estas situaciones ocurren con menos frecuencia, la fórmula de la probabilidad de supervivencia del esquema debe ser modificada de la siguiente forma:

$$p_s(E) \geq 1 - p_c \cdot \frac{ld(E)}{l-1} \quad (3.15)$$

El efecto combinado de la selección y el cruzamiento modifica la ecuación de crecimiento reproductivo del esquema:

$$I(E, t+1) \geq I(E, t) \cdot \frac{apt(E, t)}{A(t)} \left[1 - p_c \cdot \frac{ld(E)}{l-1} \right] \quad (3.16)$$

La ecuación (3.16) proporciona una cota inferior para el número esperado de cadenas representadas por el esquema E en la siguiente generación. Esta cota está en función del número real de cadenas representadas por el esquema, la aptitud relativa del esquema y su longitud definida. En esta nueva expresión, se deduce que un esquema con aptitud mayor al promedio de la generación y con longitud definida corta, mantiene una tasa de incremento exponencial.

En lo que respecta al operador mutación, se sabe que su probabilidad de ocurrencia es p_m y que su efecto es la modificación del valor de una determinada posición. Cuando un esquema sobrevive a la mutación, todas las posiciones fijas de un esquema deben permanecer sin cambio. A partir de esa reflexión, se afirma que el número de posiciones vulnerables de un esquema al operador mutación, son exactamente el número indicado por el orden del esquema.

La probabilidad de alteración de un sólo bit es p_m , entonces la probabilidad de supervivencia de un bit a la mutación simple es $1 - p_m$. La mutación de un bit es independiente de la mutación de otro bit, entonces la probabilidad de un esquema E de sobrevivir a una mutación es:

$$p_s(E) = (1 - p_m)^{\alpha(E)} \quad (3.17)$$

Como consecuencia, la forma de la ecuación de crecimiento reproductivo del esquema considerando el efecto combinado de selección, cruzamiento y mutación es:

$$I(E, t+1) \geq I(E, t) \cdot \frac{apt(E, t)}{A(t)} \left[1 - p_c \cdot \frac{ld(E)}{l-1} \right] (1 - p_m)^{\alpha(E)} \quad (3.18)$$

La ecuación 3.18 indica una cota inferior para el número esperado de cadenas representadas en el esquema E de la siguiente generación en función del número real de cadenas representadas por el esquema, la aptitud relativa del esquema, su longitud definida

y su orden. A partir de ella, también se deduce que los esquemas con promedio de aptitud mayor al promedio de aptitud de la generación, con longitud definida corta y bajo orden obtienen tasas de incremento exponencial.

Se ha observado que el efecto combinado de los operadores sobre un esquema es insignificante si el esquema es de longitud definida corta y de orden bajo. Así, el resultado final de la ecuación de crecimiento (3.18) puede ser escrito como en el siguiente teorema.

Teorema del Esquema. Un esquema superior al promedio, de longitud definida corta y de orden bajo, recibe intentos incrementados exponencialmente en las subsecuentes generaciones de un algoritmo genético.

Como resultado de este teorema se tiene que los algoritmos genéticos exploran el espacio de búsqueda por medio de esquemas de longitud definida corta y bajo orden, los cuales son utilizados subsecuentemente en el cruzamiento para intercambiar información; es decir, un algoritmo genético indaga por todo el espacio de búsqueda a través de la yuxtaposición de esquemas de corta longitud definida, bajo orden y alto desempeño.

No obstante, un algoritmo genético puede converger a una solución subóptima. Llamado *decepción*, este fenómeno ocurre cuando algunos esquemas de corta longitud definida y bajo orden, desvían al algoritmo para converger en puntos subóptimos. La utilización del conocimiento *a priori* en la función de aptitud, del operador de *inversión*, de técnicas elitistas o de los algoritmos genéticos desordenados, son soluciones potenciales de esta problemática.

4. ALGORITMOS GENÉTICOS PARA CARTERAS DE INVERSIÓN

4.1 Introducción

En los dos capítulos previos, se han presentado de forma simple e independiente los temas de carteras de inversión y algoritmos genéticos. Esos capítulos son la base del presente, en éste se describe la forma en que se han construido los algoritmos genéticos necesarios para aplicar dicha técnica de optimización a la solución del problema de creación y administración de carteras de inversión.

Mediante los cinco algoritmos genéticos que se han construido, el inversionista será capaz de crear y administrar una cartera de inversión; pero lo más importante es que podrá hacerlo tomando en cuenta el horizonte temporal, el universo de activos, la inclusión de un activo libre de riesgo y el grado de aversión al riesgo que posea.

El primer algoritmo genético pretende realizar una buena distribución estratégica de activos; es decir, su objetivo es elegir, de entre el conjunto de activos financieros riesgosos disponibles, aquellos que representen una mejor alternativa para la conformación de la cartera.

El segundo algoritmo genético, una vez que el subconjunto de activos financieros riesgosos ha sido definido, intenta encontrar una buena distribución táctica. En otras palabras, se desea conocer una buena combinación de pesos o ponderaciones que puedan ser asignados a cada activo financiero.

El tercero de ellos, se encargará de encontrar los pesos necesarios de cada activo seleccionado en el primer algoritmo, para formar la cartera de riesgo mínimo.

El cuarto, a partir del resultado otorgado por el primero y considerando el rendimiento que otorga un activo libre de riesgo, encuentra la cartera óptima de mercado;

esta cartera óptima de mercado es la misma a la que se refiere el teorema de un fondo de inversión.

El último de los algoritmos genéticos construidos, es el que permitirá al inversionista encontrar su cartera óptima, también conocida como cartera completa. En este caso se habla de su cartera óptima, no de la cartera óptima, debido a que este algoritmo genético creará una cartera de inversión considerando el grado de aversión al riesgo que presenta el inversor.

Estos cinco algoritmos genéticos son usados para la creación y administración de carteras de inversión. En conjunto, estos cinco algoritmos, representarán una herramienta muy poderosa para el inversionista, ya que le permitirán crear una nueva cartera de inversión, tan frecuentemente como él lo decida.

Para efectos de esta tesis, los algoritmos genéticos fueron programados en el paquete MATLAB; los códigos programados se encuentran en el Apéndice 1. No obstante, la lectura del texto no obliga a programar en dicho paquete, ya que sólo se explica de forma descriptiva el bosquejo general de los esquemas de los algoritmos genéticos programados. Esto permite que el lector, si lo desea, pueda programarlos en cualquier otro paquete.

4.2 Elección de los activos financieros

El primer algoritmo genético, a partir de ahora llamado algoritmo genético 1, tiene como objetivo elegir el conjunto de activos financieros riesgosos que conformarán la cartera de inversión utilizando los dos operadores genéticos de cruzamiento y mutación.

Se parte del supuesto que señala la existencia de un conjunto U de activos financieros con riesgo, $U = \{a_1, a_2, \dots, a_m\}$, en el que a_i representa al i -ésimo activo disponible para el inversionista. Entonces, el inversionista tendrá como objetivo elegir na

activos de entre m posibles para crear una cartera, por lo que tendrá ${}_m C_{na}$ combinaciones posibles para crear una cartera de inversión.

Además, se supone que el inversionista posee la serie histórica de rendimientos de cada uno los activos considerados en U y que cada serie histórica de rendimientos tiene una varianza distinta de cero.

El algoritmo genético 1, es programado como una función que requiere seis parámetros de entrada: una matriz de series históricas de rendimientos de los activos, llamada M , en la que cual la columna i corresponderá a la serie de rendimientos histórica de a_i ; el número de activos por seleccionar, na ; el número de cromosomas o individuos por considerar en cada generación, denotado por nc ; el número máximo de iteraciones a realizar por el algoritmo genético, denotado por ni ; la probabilidad de cruzamiento, pc ; y la probabilidad de mutación, pm . Como parámetros de salida, el algoritmo genético 1, devolverá tres elementos: el cromosoma más apto encontrado durante la ejecución del algoritmo; el valor de aptitud alcanzado por dicho cromosoma; y una matriz de tamaño $nc \times na$ correspondiente a la última generación del algoritmo, en la que cada renglón corresponderá a un individuo de la población y cada columna al activo que debe ser seleccionado para la conformación de la cartera de inversión.

A continuación se describe el funcionamiento del algoritmo genético 1.

4.2.1 Cálculos previos

Como se explicó antes, el algoritmo genético 1 tendrá seis parámetros de entrada, entre ellos la matriz M . El tamaño de esta matriz dependerá del número de activos disponibles, m , y de la cantidad de datos disponibles de cada serie histórica de rendimientos, llámese a esa cantidad nr . Entonces, la matriz M será de tamaño $nr \times m$, en donde cada columna representará a un activo financiero distinto. Sin embargo, para crear

una cartera de inversión el inversionista necesita conocer, por cada activo, su rendimiento, su desviación estándar y el coeficiente de correlación que posee con cada uno de los activos financieros considerados.

Es por esta razón que la función que ejecute al algoritmo genético 1, debe ser capaz en un principio manipular la matriz M para obtener cada uno de los datos requeridos y, posteriormente, poder ejecutar los pasos propios de los algoritmos genéticos.

Por lo tanto, es necesario crear tres funciones que tendrán como parámetro de entrada a la matriz M .

La primera de ellas, calculará el rendimiento que otorga de acuerdo a su serie histórica de rendimientos cada uno de los activos, por ello devolverá un vector r de m entradas, $r = (r_1, r_2, \dots, r_m)$, en el que r_i es el rendimiento promedio que otorga el activo i .

La segunda función, calculará la desviación estándar de cada activo, por lo que también devolverá un vector d de m entradas, $d = (d_1, d_2, \dots, d_m)$, en el que d_i es la desviación estándar del activo i .

La última de estas funciones, debe recibir la matriz de rendimientos M para generar una matriz cuadrada C de tamaño m , donde C_{ij} representa el coeficiente de correlación que existe entre los activos financieros ubicados en los lugares i y j .

4.2.2 Codificación de soluciones

Como se expuso en el capítulo tres, el primer paso para utilizar los algoritmos genéticos es encontrar una codificación adecuada del dominio de la función a optimizar. En este caso, las soluciones potenciales son los activos que se encuentran en el conjunto U ; por esa razón, se podrían codificar como cadenas de números binarios tal como se hizo en el ejemplo mostrado en la sección 3.2.3. Sin embargo, para este algoritmo se ha preferido

utilizar una codificación en números naturales que representa una buena alternativa para sustituir la codificación binaria.

Para observar el por qué la codificación en números naturales es, en esta ocasión, una buena alternativa para sustituir a la codificación binaria debemos considerar las características del problema.

Primero obsérvese que en el ejemplo de la sección 3.2.3, las soluciones codificadas se encuentran en el intervalo $[0,31]$. En el problema considerado en la presente sección, las soluciones por codificar están en un intervalo $[1,m]$; es decir, el límite superior es variable porque dos inversionistas pueden disponer de un número diferente de activos para conformar su cartera de inversión.

Como segundo punto, nótese que en la sección 3.2.3 las soluciones codificadas sólo representaban a un elemento en el conjunto $[0,31]$, por lo que bastaban 5 dígitos binarios para representar una solución potencial. Para el algoritmo genético 1, una solución potencial deberá representar a tantos elementos del conjunto U , como activos financieros sean requeridos para conformar la cartera de inversión.

La tercera razón de la utilización de la codificación en números naturales es consecuencia del efecto combinado de las primeras dos razones. Mientras el número de activos disponible sea mayor y el número de activos por seleccionar sea mayor, la extensión de las soluciones potenciales en codificación binaria del algoritmo genético 1, se verá incrementada de forma considerable.

Con respecto a esta última razón explicada, es necesario señalar que en ocasiones no es posible evitar que la codificación de soluciones este dada por cadenas de gran longitud. No obstante, existen casos en los que esas cadenas sumamente largas se pueden evitar; uno de ellos es el caso del algoritmo genético 1.

Una vez que se ha argumentado la preferencia por la codificación en números naturales, se explicará en qué consiste esta codificación y los cambios que introducirá en la aplicación de los operadores genéticos.

Se sabe que el conjunto de activos financieros disponibles es $U = \{a_1, a_2, \dots, a_m\}$ y que se desea seleccionar na activos para conformar la cartera de inversión. En este caso cada individuo, cromosoma o posible solución, será representada por una secuencia de m números enteros tomados en el intervalo $[1, m]$ y sólo los primeros na formarán la cartera de inversión. Así, cada cromosoma consta de m genes.

Por ejemplo, considérese $U = \{a_1, a_2, \dots, a_{10}\}$ y que se desea crear carteras de inversión integradas por 5 activos financieros. Si el algoritmo genético 1 diera como solución al cromosoma (8, 2, 10, 7, 4, 3, 6, 9, 5, 1), entonces la cartera de inversión debe ser formada sólo por los activos ubicados en los primeros 5 genes: a_8, a_2, a_{10}, a_7 y a_4 .

Siguiendo con el mismo ejemplo y de acuerdo a esta codificación de soluciones, el individuo (8, 2, 10, 7, 4, 3, 6, 9, 5, 8) no es una solución factible porque las posiciones 1 y 10 son asignadas al mismo activo. Por esta razón, las soluciones factibles de este problema son sólo el conjunto de permutaciones de m .

4.2.3 Población inicial

Siguiendo la estructura de los algoritmos genéticos, una vez que la codificación de soluciones potenciales ha sido resuelta, se procede a formar la primera generación de soluciones potenciales, es decir, se integra la población inicial.

La función que genera la población inicial únicamente necesita recibir dos parámetros: el número de cromosomas que serán creados, nc ; y el número de activos

financieros disponibles, m . A partir de los datos, esta función debe generar una matriz de tamaño $nc \times m$.

La forma en que la función genera a la matriz que contendrá a la población inicial es sumamente sencilla. Es necesario generar, por cada renglón de la matriz, una permutación de m números naturales a partir de los m activos disponibles. De esta forma cada renglón de la matriz representará a un individuo de la población.

4.2.4 Evaluación de la población

Ahora es necesario elaborar una función que permita evaluar el nivel de aptitud de cada uno de los individuos de la población. La función que permita hacer esto, recibirá 5 parámetros y devolverá sólo uno. Como parámetros de entrada estarán la matriz de la población, denotada por P ; el vector de rendimientos, r ; el vector de varianzas, d ; la matriz de coeficientes de correlación, C ; y el número de activos por seleccionar para la conformación de la cartera de inversión, na . Como valor de salida, la función debe crear un vector de evaluaciones v de nc entradas, $v = (v_1, v_2, \dots, v_{nc})$, donde v_i es la evaluación o valor de aptitud del individuo i .

En lo que se refiere a la función de aptitud que se utilizará, esta debe otorgar un nivel alto de aptitud a aquellos cromosomas que propicien, de acuerdo a sus primeros na genes, la cartera con mayor rendimiento, menor riesgo y mejor diversificación. Es decir, esta función de aptitud debe considerar, sólo de los primeros na activos, su rendimiento, su desviación estándar y sus coeficientes de correlación con respecto a los otros $na-1$ activos que conformarán la cartera.

Una posible función de aptitud que considere los datos antes mencionados y que podría servir para este algoritmo genético 1 es la siguiente:

$$F' = \sum_{i=1}^{na} \left[r_i - d_i + \sum_{j=i+1}^{na} (1 - C_{ij}) \right] \quad (4.1)$$

La expresión anterior otorgará a cada cromosoma un valor numérico de aptitud y es relativamente sencilla de programar; con ella, al aplicársela a cada cromosoma que conforma la población, se podría conformar el vector v que en cada entrada mostraría la aptitud otorgada a cada individuo por esta función.

Para evitar confusiones, es conveniente hacer notar que C_{ij} , no representa el coeficiente de correlación existente entre los activos i y j , sino que representa al coeficiente de correlación existente entre el activo que se ubica en el i -ésimo gen del cromosoma y el activo que se ubica en el j -ésimo gen del mismo cromosoma. De igual forma, d_i no representa la desviación estándar del i -ésimo activo, representa la desviación estándar del activo que se ubica en el i -ésimo gen del cromosoma considerado.

Esta función de aptitud cumpliría con el objetivo para el que se le creó, pero debe tomarse en cuenta que el valor otorgado por F' a un cromosoma podría ser no positivo.

Se observará un caso de este tipo de problemática mediante un ejemplo con valores extremos en los parámetros. Supóngase $U = \{a_1, a_2, a_3\}$ y que se crearán carteras de 2 activos. Además supóngase que $r = (0.1, -0.5, 0.3)$, $d = (0.1, 0.1, 0.1)$ y $C_{ij} = 1$ para toda pareja ij . También supóngase que el algoritmo genético 1, evaluará al cromosoma $c_3 = (2, 1, 3)$. Cuando se evalúe el cromosoma c_3 a partir de 4.1, se obtendría como valor de aptitud -0.6.

De acuerdo a la teoría de los algoritmos genéticos presentada en el capítulo tres, es necesario que la función de aptitud otorgue aptitudes positivas. Este defecto puede ser corregido. Obsérvese que los posibles valores no positivos otorgados por F' sólo pueden

estar originados al considerar rendimientos negativos, desviaciones estándar grandes o por la combinación de ambos factores. Por el contrario, los coeficientes de correlación jamás podrán crear tal efecto sobre F' .

Para evitar este tipo de situaciones es necesario calcular una cota inferior para los rendimientos y una cota superior para las desviaciones estándar. Para ello se crean dos funciones que calculen dichas cotas, las cuales serán incluidas en la función de aptitud del algoritmo genético 1.

La primera de ellas tiene como objetivo calcular la cota inferior del agregado de rendimientos. Esta función debe tener dos parámetros de entrada: el vector de rendimientos, r ; y el número de activos por seleccionar, na . Como resultado, únicamente mostrará la cota inferior calculada. Esta función procede de forma muy sencilla, a partir del vector r y del número de activos na , dicha función ordena los rendimientos de forma creciente y suma los primeros na rendimientos. Así, se conoce el menor agregado que es posible acumular a partir de los activos disponibles y se le denota por cr .

La segunda tiene como objetivo calcular la cota superior del agregado de varianzas. Esta función también debe tener dos parámetros de entrada: el vector de desviaciones estándar, d ; y el número de activos por seleccionar, na . Como resultado, únicamente mostrará la cota superior calculada. Esta función procede de modo análogo a la anterior; a partir del vector d y del número de activos na , la función ordena las desviaciones estándar de forma creciente, pero en este caso suma las últimas na . Así, se conoce el mayor agregado de desviaciones estándar que es posible acumular a partir de los activos disponibles, a este valor se le denotará por cd .

A partir de esas dos cotas calculadas y de una constante $c = 0.0001$, se tiene que la función de aptitud del algoritmo genético 1 es de la forma siguiente:

$$F = |cr| + cd + \frac{(na-1)na}{2} + c + \sum_{i=1}^{na} \left[r_i - d_i - \sum_{j=i+1}^{na} C_{ij} \right] \quad (4.2)$$

En la función F se incluye el valor absoluto de la cota inferior de los rendimientos, ya que dicha cota puede ser negativa. Además, se escribe la suma de unidades, ejecutada al considerar a los coeficientes de correlación, como una constante mas en la función. La constante c , hace que la “peor” evaluación posible sea siempre positiva.

La función F es una buena alternativa para evaluar los cromosomas de la población debido a que depende del rendimiento y de la desviación estándar de cada activo, así como de los coeficientes de correlación que existen entre cada pareja de activos posible. También debe resaltarse que F es una función positiva, por lo que ya no existe posibilidad de que algún cromosoma obtenga una aptitud no positiva.

Otra ventaja que ofrece la función de aptitud del algoritmo genético 1, es que está dada de tal forma que para cada matriz M que se considere, el algoritmo genético 1 crea una nueva función de aptitud para utilizarla en toda su ejecución.

4.2.5 Selección de población

El proceso de selección que realiza el algoritmo genético 1, es básicamente el mismo proceso que se presentó en el capítulo 3, por lo que sólo se describirá brevemente. En este paso, el esquema general del algoritmo genético no se ve afectado por las modificaciones realizadas en la codificación de las soluciones.

La selección de población es programada como una función que dependerá sólo de dos parámetros: la población actual, es decir, una matriz de $nc \times m$; y el vector de evaluaciones de los individuos de nc entradas. Como resultado, esta función genera una matriz de $nc \times m$, en donde cada renglón representa a los individuos seleccionados para conformar la población antes de aplicar los operadores genéticos.

La idea general es crear una función de probabilidad de densidad que permita más tarde crear una función de probabilidad de distribución para el conjunto de cromosomas de la población. Esto se hace justo como fue explicado en la sección 3.3 del capítulo 3. Una vez que se cuenta con la función de distribución, se generan nc números aleatorios en el rango $[0, 1]$. Para cada número aleatorio se busca el intervalo en el que se encuentra de acuerdo a la función de distribución previamente creada. De esta forma se encontrarán dos probabilidades (los límites del intervalo seleccionado) correspondientes a dos cromosomas distintos de la población, de los cuales el que corresponda al límite superior del intervalo será seleccionado para formar parte de la nueva generación a la que posteriormente se le aplicará los operadores genéticos.

4.2.6 Cruzamiento

El operador cruzamiento es ejecutado por una función que tiene dos parámetros de entrada: una matriz de tamaño $nc \times m$, con la población actual, y el valor de la probabilidad de cruzamiento pc . Como resultado, esta función debe proporcionar una nueva matriz de tamaño $nc \times m$ con la población creada después de aplicar el operador de cruzamiento.

La forma de proceder de este operador es la misma que se ha explicado antes. Para cada individuo se genera un número aleatorio en el rango $[0, 1]$ que será comparado con pc . Si el número aleatorio generado es menor que pc , se selecciona al cromosoma correspondiente para efectuarle cruzamiento, en caso contrario no es seleccionado.

Una vez que se tiene el total de cromosomas que serán alterados mediante el operador cruzamiento, debe verificarse que el número de cromosomas sea par. Cuando el número de cromosomas seleccionados es impar, se genera un número entero a aleatoriamente en el rango $[1, nc]$. Ahora se verifica, si el cromosoma a había sido

seleccionado entre los cromosomas que se cruzarían, éste se elimina de entre ellos; en caso contrario, el cromosoma a se incluye entre los cromosomas para utilizar el operador cruzamiento. De esta forma se obtendrá un número par de cromosomas.

A continuación, para cada pareja, se selecciona un número entero aleatoriamente en el rango $[1, m-1]$, que indique la posición de cruzamiento de la pareja.

Es momento de observar una de las modificaciones que provocó el haber utilizado una codificación del dominio distinta a la codificación binaria. Considérense los siguientes dos cromosomas $c_1 = (8, 2, 10, 7, 4, 3, 6, 9, 5, 1)$ y $c_2 = (5, 6, 9, 1, 4, 2, 10, 8, 3, 7)$. Ahora supóngase que el número entero aleatorio que se generó para indicar la posición de cruzamiento fue 5. Entonces aplicando el operador cruzamiento en su forma básica se obtendrían los siguientes dos cromosomas: $c_3 = (8, 2, 10, 7, 4, 2, 10, 8, 3, 7)$ y $c_4 = (5, 6, 9, 1, 4, 3, 6, 9, 5, 1)$. Se puede observar que c_3 y c_4 no representan soluciones factibles porque, en cada uno de ellos, se han designado dos posiciones a un mismo elemento.

Para que el operador cruzamiento no genere individuos que representen soluciones no factibles, se suele modificar ligeramente el operador. En este caso el primero de los hijos tendrá los mismos genes que el primero de los padres desde la posición uno hasta la posición de cruzamiento; a partir de la posición del cruzamiento, el hijo uno tendrá los genes que le hacen falta para ser una solución factible, pero respetando el orden que siguen éstos en el padre 2. Análogamente se crea al hijo dos.

De acuerdo al ejemplo anterior, $c_1 = (8, 2, 10, 7, 4, 3, 6, 9, 5, 1)$ y $c_2 = (5, 6, 9, 1, 4, 2, 10, 8, 3, 7)$ y la posición de cruzamiento es 5. Aplicando el operador cruzamiento con la modificación, los hijos que remplazarán a los padres serán $c_3 = (8, 2, 10, 7, 4, 5, 6, 9, 1, 3)$ y $c_4 = (5, 6, 9, 1, 4, 8, 2, 10, 7, 3)$. Así, los cromosomas hijos serán soluciones factibles.

4.2.7 Mutación

El operador mutación será ejecutado por una función que recibe como parámetros a la matriz de la población, después de haber aplicado el operador cruzamiento, de tamaño $nc \times m$ y a la probabilidad de mutación pm . Como resultado esta función debe dar una nueva matriz de tamaño $nc \times m$, con la población alterada por el operador mutación.

Una vez más la codificación de dominio utilizada en el algoritmo genético 1, impide que el operador en cuestión sea aplicado en la forma que ha sido explicado hasta este momento. A continuación, se explica las modificaciones que sufre el operador de mutación, para que su aplicación mantenga la factibilidad de las soluciones representadas por los individuos de la población.

Primero se genera un número aleatorio para cada gen de cada cromosoma de la población en el rango $[0, 1]$. Ahora debe compararse, para cada gen, el número aleatorio generado con pm . Si el número aleatorio es menor que pm , entonces el gen mutará; si el número aleatorio es mayor que pm , entonces el gen no mutará.

Ahora la función que ejecutará al operador mutación, debe verificar el número de genes que mutarán en cada cromosoma. En todos aquellos que exista al menos un gen que mutará, debe verificarse la condición de que el número total de genes que muten de dicho cromosoma sea par. Esta condición debe ser cumplida para que la mutación tenga sentido y mantenga la factibilidad de las soluciones representadas por los individuos de la población.

En caso de que exista algún cromosoma con un número total de genes por mutar impar, se generará un número entero a aleatorio en el rango $[1, m]$. El número generado a , indicará el gen que debe ser revisado; si a no está marcado para mutar, entonces ahora se le marcará para mutar; y si a estaba marcado para mutar, entonces se elimina la marca, de tal

suerte que ahora ya no mutará. De esta forma, en cada cromosoma existirán k genes por mutar y k será un número par.

Ahora el operador mutación, para cada pareja de genes consecutivos que mutarán en un mismo cromosoma, sólo intercambiará la posición de los genes que forman la mencionada pareja. De esta forma el operador mutación mantiene la factibilidad de las soluciones representadas por los individuos de la población.

Como ejemplo, considérese el cromosoma $c_1 = (4, 1, 5, 7, 10, 6, 9, 2, 8, 3)$ y supóngase que las posiciones 3, 6 y 7 fueron elegidas para mutar. Debido a que el número de genes por mutar es impar se genera un número entero aleatorio en el rango $[1,10]$, supóngase que dicho número fue 10. Entonces la posición 10 del cromosoma también deberá mutar. De esta forma la posición 3 se intercambiará con la 6 y la posición 7 lo hará con la 10. El cromosoma después de haber mutado es $c'_1 = (4, 1, 6, 7, 10, 5, 3, 2, 8, 9)$, mismo que representa una solución factible del problema.

4.2.8 Elitismo

A partir de lo expuesto en el capítulo anterior sobre algoritmos genéticos, es posible hacer la siguiente reflexión. La diversidad de población y precisión selectiva están fuertemente relacionados: un incremento en la precisión selectiva disminuye la diversidad de la población, y viceversa. En otras palabras, una fuerte precisión selectiva puede hacer inefectiva la búsqueda.

Los dos objetivos en el proceso de evolución de la búsqueda genética, diversidad de población y precisión selectiva, han sido resueltos mediante las implementaciones realizadas en el algoritmo genético 1. Sin embargo, el algoritmo genético 1 tendrá otra

característica que le permita alcanzar esos dos objetivos de una mejor forma; será un algoritmo genético elitista.

El elitismo es un mecanismo que se creó con el objetivo de conseguir siempre una búsqueda efectiva, independientemente de la diversidad de la población y sin importar la magnitud de la precisión selectiva que se utilice.

El elitismo es un criterio aplicable en los algoritmos genéticos, su objetivo es fortalecer la preservación del mejor cromosoma en cada generación o ciclo del algoritmo. La creación del modelo elitista se debe a DeJong quien lo publicara en 1975.

El momento de su ejecución no está definido; es decir, puede aplicarse elitismo en la selección de la población, en el paso de cruzamiento o en el de mutación e incluso al finalizar la iteración. Sin embargo, el objetivo perseguido con su ejecución no varía: privilegiar al cromosoma o al grupo de cromosomas que posean los mejores valores de aptitud mediante su inclusión en el siguiente paso del algoritmo o en la siguiente generación.

En el algoritmo genético 1, la ejecución del criterio elitista se hará al finalizar la iteración correspondiente. Para cada iteración del algoritmo, en un principio se calcula el valor de aptitud de cada individuo de la población y se crea una copia del cromosoma con el mayor valor de aptitud. A continuación, el algoritmo genético 1 ejecuta los pasos antes explicados; es decir, selecciona una población, aplica el operador de cruzamiento, aplica el operador de mutación y evalúa nuevamente la población. En este momento es cuando se ejecutará el criterio elitista; se realiza una comparación entre el valor de aptitud del cromosoma con mayor aptitud de la población al inicio de la iteración, con el valor de aptitud del cromosoma con menor aptitud de la población al finalizar la iteración. Si el primer cromosoma tiene menor valor aptitud que el segundo, entonces la población final

queda intacta; pero si ocurre lo contrario, el primer cromosoma sustituye al segundo en la población final y con esta nueva población se inicia una iteración más.

Ahora sí, el algoritmo genético 1 repetirá todos los pasos antes descritos tantas veces como lo indique el parámetro ni . Una vez que el número de iteraciones ni haya sido alcanzado, el algoritmo genético uno se detendrá para mostrar los resultados obtenidos.

Debe mencionarse que se prefirió utilizar como condición de paro el número de iteraciones ni y no la convergencia del algoritmo debido a que, como se comentó en el capítulo previo, la convergencia total de los algoritmos genéticos no siempre se da. También debe resaltarse que pensando en este posible inconveniente, es que el algoritmo genético se implementó como una función que muestra como resultados no sólo la matriz con la población final, sino también muestra al cromosoma más apto de la generación y su valor de aptitud.

4.3 Determinación de los pesos por asignar a los activos financieros

El segundo algoritmo genético, que a partir de este momento será llamado algoritmo genético 2, tiene como objetivo determinar los pesos o ponderaciones que adquirirán cada uno de los activos financieros con riesgo dentro de una posible cartera de inversión. El algoritmo genético 2, también recurrirá a los operadores genéticos de cruzamiento y mutación para alcanzar su objetivo.

En este caso, como punto de partida, se hacen tres supuestos. Primero se supone que el inversionista posee una matriz en la que cada columna es la serie histórica de los rendimientos de un activo financiero y que cada serie histórica de rendimientos tiene una varianza distinta de cero. El primer supuesto, en este momento, ya no representa ningún obstáculo para el inversionista; esto se debe a que la matriz requerida, es la misma que ha sido utilizada para la ejecución del algoritmo genético 1.

El segundo supuesto, obliga a que el inversionista haya determinado previamente los activos financieros que utilizará para conformar la cartera de inversión y que conozca la ubicación de las respectivas series históricas de los rendimientos dentro de la matriz requerida en el primer supuesto. Para satisfacer este segundo supuesto, el inversionista simplemente debió haber ejecutado previamente el algoritmo genético 1; justamente el algoritmo genético 1, determina los activos más adecuados para formar la cartera de inversión y la ubicación de cada una de las series históricas correspondientes dentro de la matriz de rendimientos.

El último supuesto indica que, en la determinación de ponderaciones de la cartera de inversión, debe considerarse que al inversionista no se le permite o no desea realizar ventas en corto. Para justificar la existencia de este supuesto, es necesario recordar lo discutido en el capítulo dos acerca de las ventas en corto. En ese capítulo, se señala que las ventas en corto son sumamente riesgosas por su pérdida potencial ilimitada; por esa razón las ventas en corto son prohibidas dentro de algunas instituciones financieras y evitadas básicamente como política por muchas otras. No obstante, es ineludible reconocer que para algunos inversionistas, éste supuesto puede representar el elemento decisivo para no utilizar el algoritmo genético 2; en otras palabras, el inversionista podría recurrir a la aplicación de alguna otra técnica o incluso a programar un algoritmo genético que considere la posibilidad de realizar ventas en corto.

Una vez que han sido discutidos los supuestos bajo los cuales operará el algoritmo genético 2, se procede a la descripción del mismo.

El algoritmo genético 2, se programa como una función que necesita seis parámetros de entrada: una matriz de series históricas de rendimientos de los activos, M , en la que la columna i corresponderá a la serie de rendimientos histórica de a_i ; un vector, u ,

que indique la ubicación de las columnas en M , correspondientes a las series de rendimientos de los activos financieros seleccionados para conformar la cartera de inversión; el número de cromosomas considerados en cada generación, nc ; el número máximo de generaciones por evolucionar en el algoritmo genético, ni ; la probabilidad de cruzamiento, pc ; y la probabilidad de mutación, pm . Como parámetros de salida, el algoritmo genético 2, otorgará tres resultados: el cromosoma más apto encontrado durante la ejecución del algoritmo; el valor de aptitud alcanzado por dicho cromosoma; y la matriz que muestra, por renglón, a cada uno de los cromosomas de la última generación del algoritmo. Nuevamente el más importante, dentro de los parámetros de salida, es el vector que mostrará al cromosoma más apto encontrado a lo largo de la ejecución del algoritmo genético 2; como es de imaginar, ese vector proporcionará los pesos más convenientes encontrados por asignarse a los activos, que constituirán la cartera de inversión que represente los objetivos planteados por la función de aptitud por utilizar.

4.3.1 Cálculos previos

El algoritmo genético 2 recibe seis parámetros de entrada; los dos primeros (M y u) son propios del problema financiero por resolver y los últimos cuatro (nc , ni , pc y pm) son propios de la técnica que se utilizará para resolverlo.

Para iniciar, debe observarse que no toda la información contenida en la matriz M es necesaria para alcanzar el objetivo que persigue este algoritmo. Para ello, nótese que la matriz M contiene las series de rendimientos históricas de todos los activos financieros que el inversionista pudo haber utilizado. Sin embargo, el vector u como resultado del algoritmo genético 1, indica los activos elegidos para formar la cartera de inversión. Como consecuencia se deben crear tres funciones que sean capaces de generar la información

necesaria sobre rendimientos, desviaciones estándar y coeficientes de correlación, pero restringiéndose a los activos seleccionados por el algoritmo genético 1.

Se debe crear una función que calcule el rendimiento promedio de las series de rendimientos históricas de cada activo. Esta función recibe dos parámetros: la matriz M de rendimientos de $n_r \times m$, donde n_r denota el número de rendimientos disponibles y m el número de activos que el inversionista pudo haber utilizado; y el vector u de n entradas, cada entrada indica la columna correspondiente al activo por incluir en la cartera. Como resultado la función debe crear un vector r de n entradas (n es el número de activos elegidos para integrar la cartera), en el que cada entrada indica el rendimiento promedio del respectivo activo. Por ejemplo, si existen 15 activos, se seleccionan 5 para formar la cartera y $u = (1,6,7,10,11)$, entonces esta primera función debe crear el vector $r = (r_1, r_6, r_7, r_{10}, r_{11})$. Recuérdese que el vector u , resultado del algoritmo genético 1, siempre mostrará el número de los activos elegidos en orden ascendente.

La función creada para calcular las desviaciones estándar recibe los parámetros M y u . Análogamente a la función anterior, crea un vector $d = (d_1, d_2, \dots, d_n)$ donde cada entrada indica la desviación estándar de los rendimientos correspondientes al respectivo activo financiero.

La tercera función también tiene como parámetros de entrada a los parámetros M y u . Con base en dichos parámetros, esta función debe crear una matriz C de $n \times n$ donde C_{ij} indica el coeficiente de correlación existente entre el activo ubicado en la entrada i y el activo ubicado en la entrada j , ambas entradas correspondientes al vector u .

Una vez que ya se han explicado las tres funciones previas, es momento de prestar atención a la codificación de soluciones y a los elementos propios de la estructura.

4.3.2 Codificación y factibilidad de soluciones

En esta ocasión, la forma de codificar las soluciones será mediante secuencias de dígitos binarios. Se sabe que la función de aptitud, aún desconocida, debe depender de los n activos financieros elegidos previamente para la conformación de la cartera. Además, se sabe que $0 \leq w_i \leq 1$ para $i = 1, 2, \dots, n$, donde w_i representa el peso asignado al i -ésimo activo seleccionado para integrar la cartera de inversión.

Conociendo los datos del problema mencionados en el párrafo anterior y suponiendo una precisión de cuatro cifras decimales en cada variable, es decir $h = 4$, se puede deducir la forma que tendrán las cadenas binarias que representarán a las soluciones potenciales del problema. Tal como se explicó en el capítulo tres, para alcanzar dicha precisión, el intervalo $[0,1]$ debe ser dividido en $(1-0) \cdot 10^4$ intervalos de igual longitud. Dado que $(1-0) \cdot 10^4 \leq 2^{14} - 1$, donde 14 es el menor número entero que hace válida la desigualdad, cada variable será representada como una cadena binaria de longitud 14. Considerando que la cartera estará formada por n activos, la representación binaria de una solución estará formada por $14 \cdot n$ dígitos binarios.

Entonces el cromosoma $v_1 = (11101010001010|10101010010001)$ representará una solución para la conformación de una cartera de dos activos; los primeros 14 dígitos representarán el peso asignado al primer activo y los últimos 14 dígitos el peso asignado al segundo activo. Ahora obsérvese lo que sucede al convertir la solución a notación decimal; aplicando la fórmula 3.1 para decodificar las cadenas binarias se obtiene:

$$w'_1 = 0 + VDC(11101010001010) \cdot \frac{1-0}{2^{14}-1} = 0 + 5207 \cdot \frac{1-0}{2^{14}-1} = 0.3178,$$
$$w'_2 = 0 + VDC(10101010010001) \cdot \frac{1-0}{2^{14}-1} = 0 + 8789 \cdot \frac{1-0}{2^{14}-1} = 0.5365.$$

Como se puede observar $0 \leq w'_1 \leq 1$ y $0 \leq w'_2 \leq 1$. Sin embargo, $w' = (w'_1, w'_2)$ no representa una solución factible porque $w'_1 + w'_2 \neq 1$.

Para convertir las soluciones binarias en soluciones factibles decimales, se debe crear una función que reciba a las soluciones en codificación binaria y que devuelva soluciones factibles en forma decimal. Para ello, esta función normalizará cada una de las soluciones que reciba, de tal forma que la suma de los pesos sea exactamente uno. Entonces, la función realizará dos operaciones para conseguir su objetivo; primero interpretará a cada solución mediante la fórmula 3.1 para encontrar w' y, como segundo paso, normalizará al vector w' para encontrar un vector w factible y en notación decimal.

Por ejemplo, si v_1 es la solución binaria, la función encontrará en primera instancia a w' y en seguida a $w = \left(\frac{0.3178}{0.3178 + 0.5365}, \frac{0.5365}{0.3178 + 0.5365} \right) = (0.3720, 0.6280)$.

4.3.3 Población inicial

La función encargada de crear la población inicial dependerá de dos parámetros, el número de cromosomas y el número de activos que se eligieron para integrar la cartera. El primero de los parámetros está dado desde un principio por nc y el segundo parámetro, que ha sido denotado antes con n , puede ser adquirido a partir de la longitud del vector u . A partir de ellos, la función creará una matriz de $nc \times 14 \cdot n$ dígitos binarios generados aleatoriamente. En la matriz, cada renglón representa a un cromosoma de la población; en cada renglón, las primeras 14 entradas representarán al primer activo, las siguientes 14 al segundo y así sucesivamente.

Nótese que la población inicial en este algoritmo genético 2, no está obligada a ser una población de soluciones potenciales factibles, sólo se puede afirmar que son soluciones potenciales en codificación binaria.

4.3.4 Evaluación de la población

La función programada para calcular la aptitud de cada individuo considerado en la población del algoritmo genético 2, recibirá 4 datos: una matriz de $nc \times n$ con elementos en notación decimal y donde cada renglón representa una solución factible (resultado de la conversión de la matriz de la población binaria); el vector de rendimientos promedio r ; el vector de desviaciones estándar d ; y la matriz de los coeficientes de correlación C . Como resultado, la función generará un vector v de n entradas, donde la i -ésima entrada indica la aptitud alcanzada por el i -ésimo cromosoma de la población.

La función de aptitud que permitirá diferenciar entre las buenas y las malas soluciones potenciales factibles está basada en los criterios de varianza mínima y del inversionista insaciable. A partir de esos dos criterios, explicados en el capítulo 2, se concluye que el inversionista busca, por un lado, carteras con menor riesgo y, por otro, carteras con mayor rendimiento. De acuerdo a lo comentado, la función siguiente podría ser una opción para calcular la aptitud de los cromosomas de la población:

$$G' = \frac{\sum_{i=1}^n w_i r_i}{\sum_{i,j=1}^n w_i w_j C_{ij} d_i d_j} \quad (4.3)$$

La ecuación (4.3), escrita de acuerdo a la notación que se ha utilizado a lo largo de este capítulo, permite calcular la proporción que existe entre el rendimiento de la cartera y el riesgo asumido al cuadrado. De esa forma es posible reflejar los criterios de varianza mínima y del inversionista insaciable en una sola ecuación. La evaluación de dos carteras

de inversión con el mismo riesgo y distinto rendimiento mediante la ecuación G' , otorgará una mayor aptitud a la cartera con mayor rendimiento. La evaluación de dos carteras de inversión con el mismo rendimiento y distinto riesgo por medio de la ecuación G' , otorgará una mayor aptitud a la cartera con menor riesgo.

Además, obsérvese que el cociente calculado en la ecuación (4.3) siempre está definido. Para comprobar esta afirmación se debe recordar que la única forma en que una cartera puede otorgar un rendimiento igual a cero es cuando esté conformada por dos activos cuyo coeficiente de correlación sea -1 , situación sumamente difícil de encontrar en el mercado. Fuera de ese caso excepcional, una cartera constituida por activos con riesgo resultado de la ejecución del algoritmo genético 1, como se explicó en el capítulo 2, al menos obligará a asumir un riesgo idéntico al riesgo sistemático o de mercado, sin importar que tan buen proceso de diversificación se haya realizado.

A pesar de todo, G' podría no ser adecuada para utilizarse como función de aptitud del algoritmo genético 2; la razón de esa posibilidad es que G' podría generar aptitudes no positivas. Para resolver esta problemática, se procederá como en la situación similar encontrada en la construcción de la función de aptitud del algoritmo genético 1. Se sabe que las posibles aptitudes no positivas sólo pueden estar originadas por la existencia de un numerador no positivo. Para evitar esa situación, se crea una función que calculará el peor rendimiento posible de la cartera. Esta función dependerá únicamente del vector de rendimientos y devolverá el valor de la constante, c , necesaria para impedir esta situación. Para encontrar dicho valor, la función ubicará el peor rendimiento, cr , entre los rendimientos contenidos en el vector r y verificará su valor: si $cr > 0$, entonces $c = 0$; en caso contrario, $c = 0.0001 - cr$. Como consecuencia, la forma de la nueva función de aptitud será:

$$G = \frac{c + \sum_{i=1}^n w_i r_i}{\sum_{i,j=1}^n w_i w_j C_{ij} d_i d_j} \quad (4.4)$$

Mediante la función G se podrán asignar valores de aptitud positivos a cada individuo de la población. Obsérvese que en el caso donde $c = 0$, la función $G = G'$, entonces se está calculando estrictamente la proporción que existe entre el rendimiento de la cartera y el riesgo asumido al cuadrado; pero cuando $c \neq 0$, la proporción calculada presenta un sesgo que dependerá de la magnitud del peor rendimiento encontrado en r .

También es importante remarcar que la forma adquirida por la ecuación (4.4) en la ejecución del algoritmo genético 2, será la misma durante todas las iteraciones; pero para dos ejecuciones con vectores $u_1 \neq u_2$, las respectivas funciones de aptitud podrían ser distintas, debido a que el cálculo de los vectores de rendimientos dependerá de u_1 y u_2 .

4.3.5 Selección de población

La función que selecciona a los individuos que podrán pasar a la siguiente generación, por sí mismos o por medio de un descendiente, necesita conocer la matriz de la población en codificación binaria y el vector de aptitudes de los individuos.

El procedimiento de selección es el mismo que se utilizó en el algoritmo genético 1 y que previamente se explicó en el capítulo 3: 1) se calcula la aptitud de cada cromosoma, 2) se encuentra la aptitud total de la población, 3) se calcula la probabilidad de selección p_i para cada cromosoma y 4) se calcula la probabilidad acumulativa q_i .

A partir de las probabilidades acumulativas calculadas y de una serie R_g de nc números generados aleatoriamente en el intervalo $[0,1]$, se procede a realizar las respectivas comparaciones que permitan encontrar nc individuos para integrar la nueva población que

se verá afectada por los operadores genéticos. Finalmente, la función crea una matriz de $nc \times 14 \cdot n$ con los individuos que fueron seleccionados, es decir, es una matriz donde cada renglón representa a un individuo en codificación binaria.

4.3.6 Cruzamiento

La función que aplica el operador de cruzamiento será ejecutada a partir de la matriz de $nc \times 14 \cdot n$ creada en el paso de selección y de la probabilidad de cruzamiento dada por el parámetro pc , introducida por el usuario desde el inicio de la ejecución del algoritmo genético 2.

Una vez más, el operador cruzamiento será usado sobre dos cromosomas de la población a partir de un punto de cruce. En este caso, el operador de cruzamiento se ejecuta tal y como fue explicado en el capítulo 3. A partir de una serie R_c de nc números generados aleatoriamente en el intervalo $[0,1]$, se procede a realizar las comparaciones que determinen si el cromosoma es seleccionado para ser afectado por el operador cruzamiento o no. Se debe verificar que el número total de cromosomas por ser afectados, sea un número par, de modo que puedan formarse las parejas de individuos que serán cruzados. En seguida, para cada pareja, se elige el punto de cruce y se realiza el cruzamiento.

4.3.7 Mutación

La función que aplica el operador mutación tiene como parámetros de entrada una matriz de $nc \times 14 \cdot n$ que contiene a la población y la probabilidad de cruzamiento. Como resultado, la función proporcionará una nueva matriz de $nc \times 14 \cdot n$ con la población alterada y aún codificada en forma de cadenas binarias.

La ejecución del operador mutación se realiza exactamente como se explicó en el capítulo 3, es decir, bit a bit y con una serie R_m de $nc \times 14 \cdot n$ números generados

aleatoriamente en el rango $[0,1]$. Los números de la serie R_M servirán para realizar las comparaciones necesarias que indicarán si el bit en cuestión debe mutar o no.

4.3.8 Elitismo

El algoritmo genético 2, como el algoritmo genético 1, será un algoritmo elitista. Para cada generación, al iniciar el proceso evolutivo se calcula el valor de aptitud de cada individuo de la población y se crea una copia del cromosoma con el mayor valor de aptitud. A partir de este momento, el algoritmo genético 2 selecciona una población y aplica los operadores de cruzamiento y mutación.

Cuando la población ya ha mutado, ésta debe convertirse en una población de soluciones factibles y en representación decimal para calcular la aptitud de cada individuo. En seguida, se ejecuta el criterio elitista mediante una comparación entre la aptitud del cromosoma copiado al inicio de la iteración y el valor de aptitud del cromosoma con menor aptitud al finalizar la iteración. Si el primer individuo tiene una aptitud menor que el segundo, entonces la población final queda intacta; en caso contrario, el primer cromosoma ocupa el lugar del segundo en la población final para iniciar una iteración y continuar de la misma forma hasta alcanzar las n_i iteraciones.

4.4 Determinación del punto de varianza mínima

Mediante el algoritmo genético 1, se definió un subconjunto de activos riesgosos que representa una buena opción para, a partir de ellos, crear una cartera de inversión diversificada. Con el algoritmo genético 2, se ubicaron los pesos que deberían ser asignados a cada activo considerado, para formar la cartera de inversión que mostrará el mejor cociente entre el rendimiento esperado y el riesgo que representa su posesión para el inversionista.

Ahora con el algoritmo genético 3, el inversionista será capaz de definir los pesos que deben ser asignados a cada activo riesgoso, para construir la cartera de inversión que está representada por el punto de varianza mínima en el diagrama riesgo – rentabilidad generado a partir de los activos considerados por el inversionista.

4.4.1 Características generales del algoritmo genético 3

El algoritmo genético 3 se desprende del anterior; para justificar esta afirmación es conveniente observar las características del problema que ahora se desea resolver. En este caso, ya se conocen los activos financieros riesgosos que deben ser incluidos en la cartera de inversión y no se conocen las ponderaciones que deben recibir cada uno de ellos. Se observa fácilmente que las características del nuevo problema son iguales a las del problema resuelto mediante el algoritmo genético 2.

A partir de la observación del párrafo anterior, es justificable que el algoritmo genético 3 tenga una misma forma para codificar soluciones, generar la población inicial, seleccionar la población, aplicar los operadores genéticos y aplicar el elitismo. El cambio que implica la existencia de un nuevo algoritmo genético, se encuentra en la función que sea capaz de evaluar las soluciones potenciales de este nuevo problema. Por este motivo, sólo la función de evaluación será explicada como antes, pero no así los otros elementos de la estructura del algoritmo genético 3, debido a que éstos son programados como se hizo en el algoritmo genético 2.

El algoritmo genético 3 recibe seis parámetros de entrada; los dos primeros (M y u) son propios del problema financiero por resolver y los últimos cuatro (nc , ni , pc y pm) son propios de la técnica que se utilizará para resolverlo. Los seis parámetros, como en el algoritmo genético 2, se interpretan de la siguiente forma: M es la matriz de rendimientos del conjunto total de activos financieros riesgosos disponibles por el inversionista, u es el

vector que indica los activos que serán incluidos en la cartera de inversión, nc es el número de cromosomas o individuos por considerar en cada una de las ni iteraciones que se desean ejecutar, pc es la probabilidad de que un cromosoma sea seleccionado para aplicarle el operador cruzamiento y pm es la probabilidad de que un gen de un cromosoma sea seleccionado para aplicarle el operador mutación.

Al finalizar su ejecución, el algoritmo genético 3 mostrará un vector que contendrá las ponderaciones necesarias para formar la cartera de inversión buscada, el valor de aptitud alcanzado por el cromosoma representado por el vector y una matriz que muestra a los individuos que integraron la última generación del algoritmo genético 3.

4.4.2 Evaluación de la población

Como se ha mencionado anteriormente, el algoritmo genético 3 pretende encontrar la cartera de inversión que es representada por el punto de varianza mínima en un diagrama riesgo – rentabilidad.

En este caso la función de evaluación de la población, deberá otorgar una aptitud mayor a los cromosomas que representen carteras de inversión con riesgo bajo; por el contrario, deberá otorgar aptitudes bajas a las soluciones potenciales que definan una cartera de inversión con riesgo alto.

Originalmente la función por optimizar sería la expresión que define la desviación estándar de una cartera de inversión; sin embargo, esta expresión implicaría que la función objetivo debería ser minimizada. Como se explicó durante el capítulo anterior, utilizando los algoritmos genéticos como una técnica de optimización, se acostumbra escribir los problemas en términos de maximización; para lograr este cambio sólo necesita considerarse una constante suficientemente grande que al restarle la desviación estándar de cualquier cartera posible, asegure que la aptitud se conserve positiva.

A partir del algoritmo genético 1, se sabe qué activos son los que integrarán la cartera de inversión. A partir de esos activos, el algoritmo genético 3 deberá ubicar la desviación estándar más grande que entre ellos se presenta; así la constante c que se utilizará en la función de aptitud será igual a la mayor desviación estándar encontrada entre los activos considerados más 0.0001. De esta forma, la función de aptitud del algoritmo genético 3 es:

$$G = c - \sqrt{\sum_{i,j=1}^n w_i w_j C_{ij} d_i d_j} \quad (4.5)$$

Con la ecuación (4.5), cualquier solución potencial obtendrá una aptitud positiva, el peor de los casos se encuentra cuando se evalúe la cartera compuesta sólo por el activo que presentó la mayor desviación estándar, en esta situación su valor de aptitud será de 0.0001. Por el contrario, el mejor de los casos se presentará cuando se evalúe a la solución que representa la cartera de varianza mínima; aquí, la aptitud alcanzará el máximo valor posible de G .

Para finalizar con este tercer algoritmo genético, vale la pena hacer notar que la función que calculaba los rendimientos diarios promedio, ya no es necesaria en este algoritmo. Esto se debe a que la función de aptitud únicamente depende de las desviaciones estándar y los coeficientes de correlación involucrados con los activos financieros con riesgo que se consideran para la formación de la cartera de inversión.

4.5 Inclusión del activo libre de riesgo

Mediante los algoritmos genéticos 1, 2 y 3, el inversionista es capaz de seleccionar los activos financieros riesgosos y encontrar las ponderaciones que deben asignárseles a cada uno, para obtener la cartera de inversión con el mejor cociente rendimiento – riesgo y la de varianza mínima. Sin embargo, no se ha tomado en cuenta que

el inversionista tiene la posibilidad de invertir en un activo libre de riesgo. Al considerar este nuevo aspecto, como se explicó en el capítulo 2, la forma de la frontera eficiente se modifica para convertirse en una línea recta que es tangente a la frontera eficiente que se tenía en un principio; justo el punto de tangencia involucrado, es la cartera de inversión de activos riesgosos mencionada en el Teorema de un Fondo.

Por medio del algoritmo genético 4, el inversionista será capaz de encontrar las ponderaciones que deben ser asignadas a los activos riesgosos, para formar la cartera de inversión que es mencionada en el teorema de un fondo.

4.5.1 Características generales del algoritmo genético 4

El algoritmo genético 4 también se desprende del segundo. En este caso, también se conocen los activos financieros riesgosos que deben ser incluidos en la cartera de inversión, no se conocen las ponderaciones que deben recibir cada uno de ellos y se sabe que existe un activo libre de riesgo. Se observa fácilmente que a excepción de la existencia del activo libre de riesgo, las características del nuevo problema son casi las mismas a las del segundo.

El algoritmo genético 4 tiene una misma forma para codificar soluciones, generar la población inicial, seleccionar la población, aplicar los operadores genéticos y aplicar el elitismo que el segundo.

El algoritmo genético 4 recibe siete parámetros de entrada; los tres primeros (M , u y lr) son propios del problema financiero por resolver y los últimos cuatro (nc , ni , pc y pm) son propios de la técnica que se utilizará para resolverlo. El único parámetro de entrada nuevo, es precisamente lr mismo que denota al rendimiento otorgado por el activo libre de riesgo; este parámetro nuevo debe ser escrito en notación decimal. Los seis parámetros restantes, como en los algoritmos genéticos 2 y 3, se interpretan de la forma siguiente: M es

la matriz de rendimientos del conjunto total de activos financieros riesgosos disponibles por el inversionista, u es el vector que indica los activos que serán incluidos en la cartera de inversión, nc es el número de cromosomas o individuos por considerar en cada una de las ni iteraciones que se desean ejecutar, pc es la probabilidad de que un cromosoma sea seleccionado para aplicarle el operador cruzamiento y pm es la probabilidad de que un gen de un cromosoma sea seleccionado para aplicarle el operador mutación.

Al finalizar su ejecución, el algoritmo genético 4 mostrará un vector que contendrá las ponderaciones necesarias para formar la cartera de inversión buscada, el valor de aptitud alcanzado por el cromosoma representado por el vector y una matriz que muestra a los individuos resultado de la última generación del algoritmo genético 4.

4.5.2 Evaluación de la población

Como se mencionó anteriormente, el algoritmo genético 4 pretende encontrar la cartera de inversión que representa el fondo único al que se refiere el teorema del mismo nombre. Gráficamente, en un diagrama riesgo – rentabilidad, el punto que representa a la cartera de inversión buscada y el punto que representa al activo libre de riesgo, definen una recta cuya pendiente puede ser calculada. Ahora bien, si a partir de cada cartera ubicada en la región factible generada por los activos con riesgo, se define la línea recta que pasa por el punto que representa a dicha cartera y el punto que representa al activo libre de riesgo, obtendríamos una infinidad de rectas de las cuales sólo la que tenga la mayor pendiente sería de interés para el inversionista. Así, la recta con mayor pendiente será la recta tangente que estará definida por el activo libre de riesgo y la cartera buscada.

La pendiente de la recta definida por el punto que representa al activo libre de riesgo y el punto que representa a una cartera en un diagrama de riesgo – rentabilidad, se puede calcular como:

$$G' = \frac{\sum_{i=1}^n w_i r_i - lr}{\sqrt{\sum_{i,j=1}^n w_i w_j C_{ij} d_i d_j}} \quad (4.6)$$

La ecuación (4.6) es una opción para ocupar el lugar de la función de aptitud del algoritmo genético 4; sin embargo, como antes, debemos evitar que la función de aptitud otorgue valores no positivos a los individuos de la población. Para que la función de aptitud siempre otorgue valores positivos a las posibles soluciones, se define la nueva función de aptitud como:

$$G = \begin{cases} \frac{\sum_{i=1}^n w_i r_i - lr}{\sqrt{\sum_{i,j=1}^n w_i w_j C_{ij} d_i d_j}} & \text{si } \sum_{i=1}^n w_i r_i - lr > 0 \\ 0.0001 & \text{si } \sum_{i=1}^n w_i r_i - lr \leq 0 \end{cases} \quad (4.7)$$

Mediante la expresión (4.7) se tiene la función de aptitud que permitirá encontrar al algoritmo genético 4, el punto que representa a la cartera de inversión que junto con el punto que representa al activo libre de riesgo, definen la recta tangente buscada.

4.6 Inclusión de la aversión al riesgo del inversionista

Con el resultado que proporciona el algoritmo genético 4, el inversionista es capaz de definir la frontera eficiente definida por los activos con riesgo y el activo libre de riesgo. Es posible que la cartera encontrada como el punto de tangencia, sea la cartera óptima del inversionista; pero también es posible que no sea así. Para saber si esto es cierto, se debe considerar el grado de aversión al riesgo que presenta el inversionista.

El grado de aversión al riesgo, quedará definido por un índice A , donde $A \geq 0$. Dado que el valor de A define la pendiente de la curva de utilidad del inversionista, se

deduce que a mayor valor de A , mayor grado de aversión al riesgo; es decir, un inversionista con un índice A elevado, solicita incrementos grandes en rentabilidad para un incremento unitario de riesgo.

A partir de la consideración de la aversión del riesgo del inversionista, el problema a resolver es maximizar el valor de la función de utilidad del inversionista sujeto a las carteras de inversión que integran la recta tangente que representa la frontera eficiente. Gráficamente, el problema es encontrar el punto sobre la frontera eficiente que sea tangente a una de las curvas de utilidad del inversionista. Este es el problema que el algoritmo genético 5 se encargará de resolver.

4.6.1 Características generales del algoritmo genético 5

El algoritmo genético 5 busca encontrar un valor y , donde $0 \leq y \leq 1$, tal que el punto $(\sigma_p, R_p) = (1 - y) \cdot (0, I_r) + y \cdot (\sigma_c, R_c)$ es tangente a una de las curvas de utilidad del inversionista. En esta ocasión, los valores de entrada necesarios son diferentes a los que se utilizaron en los tres algoritmos genéticos anteriores; no obstante, la codificación de soluciones potenciales es, una vez más, la codificación binaria.

Dado que la codificación binaria será nuevamente utilizada, el algoritmo genético 5 genera la población inicial, selecciona la población, aplica los operadores genéticos y aplica el elitismo, de forma similar a los algoritmos genéticos 2, 3 y 4.

El algoritmo genético 5 recibe ocho parámetros de entrada: el rendimiento esperado de la cartera que se encontró como resultado del algoritmo genético 4, denotado por r_p ; la desviación estándar de la misma cartera, d_p ; el rendimiento otorgado por el activo libre de riesgo, I_r ; el índice de aversión al riesgo del inversionista, α ; el número de cromosomas por considerar en cada generación, nc ; el número de iteraciones por ejecutar,

ni ; la probabilidad de que un cromosoma sea seleccionado para aplicarle el operador cruzamiento, pc ; y la probabilidad de que un gen sea seleccionado para aplicarle el operador de mutación, pm .

Como valores de salida, el algoritmo genético 5 mostrará el valor y correspondiente al mejor cromosoma encontrado en su ejecución, el valor de aptitud alcanzado por dicho valor y una matriz de $1 \times nc$ correspondiente a la última generación del algoritmo genético.

4.6.2 Evaluación de la población

Como se explicó antes, el problema a resolver es maximizar el valor de la función de utilidad del inversor sujeto a las posibilidades de carteras de inversión que ofrece la frontera eficiente construida a partir de los activos con riesgo y el activo libre de riesgo.

La función que se utiliza para definir la función de utilidad del inversionista es:

$$U = R - \frac{A\sigma^2}{2} \quad (4.8)$$

donde R representa el rendimiento esperado de la cartera de inversión, A el índice de aversión al riesgo del inversionista y σ^2 la varianza de la cartera de inversión.

A partir de la expresión (4.8), se observa que para cada valor de U queda definida una curva de utilidad del inversionista; es entre estas curvas que se buscará la que sea tangente a la frontera eficiente.

Con la función de utilidad del inversionista definida y siendo C la cartera de inversión buscada, se puede reescribir el problema en cuestión de la forma siguiente:

$$\text{Maximizar} \quad U = R_c - \frac{A\sigma_c^2}{2} \quad (4.9)$$

sujeto a
$$R_c = lr + \sigma_c \frac{rp - lr}{dp} \quad (4.10)$$

Sustituyendo (4.10) en (4.9) y haciendo el cambio de variable $y = \frac{\sigma_c}{dp}$, la función

a maximizar sería:

$$U = rp \cdot y + lr \cdot (1 - y) - \frac{A \cdot y^2 \cdot dp^2}{2} \quad (4.11)$$

A partir de la ecuación (4.11) y considerando la restricción de que la función de evaluación del algoritmo genético 5 no puede otorgar valores de aptitud no positivos, se define la función de aptitud de este algoritmo como:

$$G = \begin{cases} rp \cdot y + lr \cdot (1 - y) - \frac{A \cdot y^2 \cdot dp^2}{2} & \text{si } rp \cdot y + lr \cdot (1 - y) > \frac{A \cdot y^2 \cdot dp^2}{2} \\ \frac{lr}{1000} & \text{si } rp \cdot y + lr \cdot (1 - y) \leq \frac{A \cdot y^2 \cdot dp^2}{2} \end{cases} \quad (4.12)$$

A partir de la función, el algoritmo genético 5 será capaz de encontrar el valor y , con $0 \leq y \leq 1$, tal que el punto sobre la frontera eficiente que representa a la cartera de inversión con rendimiento $rp \cdot y + lr \cdot (1 - y)$ y desviación estándar $y \cdot dp$, es tangente a una de las curvas de utilidad del inversionista, las cuales son definidas por su grado de aversión al riesgo A .

4.7 Administración de la cartera de inversión

Como se presentó al inicio del capítulo, los algoritmos genéticos construidos deben permitir crear y administrar carteras de inversión. Hasta el momento, con los algoritmos genéticos 1, 2, 3, 4 y 5, el inversionista es capaz de realizar la distribución estratégica y la distribución táctica de activos financieros, considerando además la inclusión de un activo libre de riesgo y su grado de aversión al riesgo.

Para entender cómo esos mismos algoritmos genéticos se utilizarán para llevar a cabo el proceso de administración de la cartera de inversión, se supone que el inversionista definió el horizonte temporal y la rotación de activos que aplicará a la cartera de inversión.

El horizonte temporal indicará el número de periodos (días, meses o años) durante los cuales se administrará la cartera. Entonces el algoritmo genético 1, se ejecutará una vez durante el periodo definido por el horizonte temporal. De esta forma, para cada horizonte temporal, se considerarán nuevamente el total de activos financieros riesgosos disponibles para el inversionista, para formar una nueva cartera de inversión.

La rotación de activos indicará el número de veces que se pueden vender y comprar los activos de la cartera a lo largo del horizonte temporal definido. Entonces los algoritmos genéticos 2, 3, 4 y 5 (según el objetivo que persiga el inversionista), se ejecutarán tantas veces como lo indique la rotación definida por el inversionista. Como consecuencia, los pesos asignados a cada activo financiero que integra la cartera, serán calculados tantas veces como lo señale la rotación.

En cuanto a los datos necesarios para ejecutar los cinco algoritmos genéticos, únicamente es necesario tomar en cuenta que la matriz de rendimientos de los activos financieros, debe ser actualizada antes de cada ejecución de los algoritmos. El objetivo de esta acción, es contar con la cantidad mayor de datos disponibles en el mercado, para intentar reflejar el comportamiento de los activos financieros de la mejor forma posible.

5. EJECUCIÓN DE LOS ALGORITMOS GENÉTICOS

5.1 Introducción

En este capítulo final, los objetivos que se persiguen son mostrar la forma en que los algoritmos genéticos programados son ejecutados y analizar los resultados que se obtienen a partir de dichas ejecuciones; todo ello con la finalidad de sustentar las conclusiones que ponen punto final a este trabajo.

Para cumplir con dichos objetivos, en primera instancia se realiza un acercamiento a los algoritmos genéticos programados, en el que se desea observar los efectos causados sobre los resultados obtenidos con motivo de la variación de los parámetros de entrada. Para ello se distingue, como se ha hecho en el capítulo anterior, entre parámetros de entrada propios del algoritmo genético (número de cromosomas, número de iteraciones, probabilidad de cruzamiento y probabilidad de mutación) y parámetros de entrada propios del problema financiero estudiado.

Específicamente, el análisis sobre los parámetros sólo se realizará considerando variaciones en aquellos que son propios del algoritmo genético y no en los que provienen del problema financiero por resolver. La razón por la que se decidió no realizar este análisis para los parámetros financieros, es que cada inversionista interesado en utilizar los algoritmos genéticos programados, podría tener una base de datos distinta a la de cualquier otro; es decir, en la mayoría de las ocasiones no todos los usuarios considerarán los mismos activos en su universo de activos disponibles, ni estarán buscando siempre formar carteras de un número determinado de activos, ni dispondrán del mismo número de rendimientos por activo. A partir de esa diversidad, resulta casi innecesario y poco importante analizar dicho aspecto.

A diferencia del análisis sobre parámetros financieros, analizar las variaciones en los resultados causadas por la modificación de los parámetros genéticos, sí es un aspecto de interés común para todos los usuarios de los algoritmos genéticos programados. Una vez que se ha realizado el análisis sobre los parámetros genéticos de entrada de los algoritmos y sus efectos al ser variados, cualquier inversionista tendrá una mejor idea de los valores convenientes por asignar a dichos parámetros o al menos sabrá que valores no son convenientes para asignárseles.

En particular para este trabajo, los resultados obtenidos en el análisis previamente comentado, formarán la base para asignar los valores a los parámetros genéticos que posibilitarán la ejecución de un par de ejemplos de creación de una cartera de inversión, de los cuales uno de ellos será extendido a la administración de la misma.

La razón por la cual el primero de los ejemplos no será continuado hasta la administración de la cartera, se debe a que en él se supone un universo de activos disponibles reducido; situación que se deja de lado en el segundo ejemplo. Este supuesto se asume con la intención de mostrar, de forma detallada, los resultados que otorgan los algoritmos genéticos para la creación de la cartera de inversión; al considerar un universo de activos disponibles extenso, sería muy difícil lograr este propósito.

En el segundo ejemplo, se crea una cartera de inversión con un universo de activos disponibles extenso, creación que es explicada con menos detalle a la anterior, y se le administra a lo largo de un periodo de tiempo determinado.

Con base en el análisis de los resultados obtenidos, tanto en la variación de parámetros de entrada genéticos, como en el par de ejemplos realizados, posteriormente se dedica una sección para plasmar las conclusiones finales obtenidas a lo largo de este trabajo.

5.2 Alteración de parámetros genéticos

En un algoritmo genético, el usuario debe definir los valores que serán asignados a cada uno de los parámetros de entrada; muchas veces esta selección de parámetros define la calidad de las soluciones encontradas en la ejecución del algoritmo. Por ejemplo, en un problema que tiene pocas soluciones factibles, puede bastar con un número pequeño de cromosomas para encontrar una solución buena al problema; sin embargo, en un problema con un espacio de búsqueda grande, el mismo número de cromosomas puede ser insuficiente para que el algoritmo encuentre soluciones buenas.

Saber exactamente que valores deben ser asignados a los parámetros de entrada para encontrar soluciones buenas es imposible, debido a la gran cantidad de decisiones que se toman basándose en la aleatoriedad en un algoritmo genético. No obstante, una forma de poder vislumbrar que valores son convenientes para asignárseles a los parámetros y que valores no lo son, es ejecutar los algoritmos variando los parámetros y analizar los resultados que estos otorgan. Esto es justamente lo que se hará en la presente sección; se ejecutarán los algoritmos genéticos variando una a uno sus parámetros en cada ejecución.

Como se recordará, cada uno de los algoritmos genéticos programados en el capítulo anterior tiene cuatro parámetros de entrada propios de los algoritmos genéticos: el número de cromosomas, nc ; el número de iteraciones por realizar, ni ; la probabilidad de aplicación del operador cruzamiento, pc ; y la probabilidad de aplicación del operador mutación, pm . La forma de analizar los efectos provocados en los resultados a partir de la variación de los parámetros, será modificando en cada ejecución a uno de ellos. Además, como punto de partida se consideran 100 cromosomas para cada una de las 100 iteraciones por realizar y las probabilidades de cruzamiento y mutación consideradas son 0.25 y 0.01, respectivamente.

Es conveniente hacer notar que el análisis propuesto, también puede realizarse variando los parámetros por parejas, tercias o en cuartetos; sin embargo, para efectos de este trabajo sólo se realizará como fue señalado en el párrafo anterior.

5.2.1 Efectos en el algoritmo genético 1

Para analizar al primer algoritmo genético, se utiliza una matriz integrada por 54 activos, cada serie compuesta por 220 rendimientos diarios del correspondiente activo y se desea seleccionar a 15 de ellos para integrar la cartera de inversión.

El algoritmo genético 1 se ejecutó 200 veces; 50 veces por cada parámetro propio del algoritmo genético. Los rangos considerados para variar los parámetros fueron: $10 \leq nc \leq 500$, $10 \leq ni \leq 500$, $0.01 \leq pc \leq 0.50$ y $0.005 \leq pm \leq 0.25$. A continuación se presentan los cinco mejores resultados obtenidos al variar cada parámetro considerado.

nc	ni	pc	pm	Cromosoma	Aptitud	Tiempo (s)
220	100	0.25	0.01	6,11,15,18,22,31,33,37,39,40,42,43,49,51,52	106.563	81.109
360	100	0.25	0.01	6,8,11,15,18,30,31,39,40,41,42,43,49,51,52	106.5236	167.750
190	100	0.25	0.01	6,11,15,18,22,30,31,33,37,39,40,42,43,49,52	106.5224	65.953
440	100	0.25	0.01	6,8,11,15,18,26,29,31,37,39,40,42,43,49,52	106.4099	229.031
350	100	0.25	0.01	6,8,11,15,25,29,31,33,37,39,40,42,43,49,52	106.3389	160.484

Tabla 5.1 Mejores resultados al variar el número de cromosomas.

nc	ni	pc	pm	Cromosoma	Aptitud	Tiempo (s)
100	310	0.25	0.01	6,8,11,15,18,22,29,31,37,39,40,42,43,49,52	106.7762	87.657
100	450	0.25	0.01	6,11,15,18,22,29,31,33,37,39,40,42,43,49,52	106.7567	127.015
100	480	0.25	0.01	6,11,15,18,22,29,31,37,39,40,42,43,49,51,52	106.7526	134.891
100	460	0.25	0.01	6,11,15,18,29,31,37,39,40,41,42,43,49,51,52	106.7225	129.532
100	360	0.25	0.01	6,8,11,15,18,29,31,37,39,40,42,43,49,51,52	106.6555	101.922

Tabla 5.2 Mejores resultados al variar el número de iteraciones.

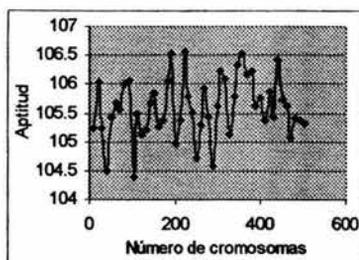
nc	ni	pc	pm	Cromosoma	Aptitud	Tiempo (s)
100	100	0.39	0.01	11,15,18,20,22,24,29,31,37,39,40,42,43,49,52	106.3826	29.860
100	100	0.02	0.01	8,11,15,18,22,25,29,33,39,40,41,42,43,49,52	106.3772	26.609
100	100	0.31	0.01	6,8,11,15,18,22,29,33,37,39,40,42,49,51,52	106.2875	29.188
100	100	0.33	0.01	6,8,11,18,25,29,31,33,37,39,40,43,49,51,52	106.2771	29.328
100	100	0.29	0.01	8,11,15,18,24,25,29,31,39,40,41,42,43,49,52	106.2739	29.094

Tabla 5.3 Mejores resultados al variar la probabilidad de cruzamiento.

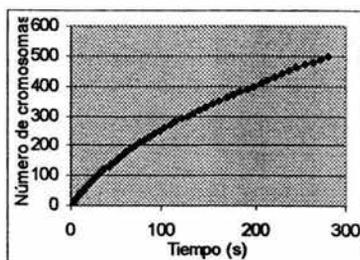
nc	ni	pc	pm	Cromosoma	Aptitud	Tiempo (s)
100	100	0.25	0.095	6,8,11,15,29,31,33,37,39,40,42,43,49,51,52	106.3266	30.266
100	100	0.25	0.190	6,8,11,15,18,29,36,37,39,40,41,42,43,49,52	106.2928	30.547
100	100	0.25	0.160	6,8,11,15,18,29,38,39,40,41,42,43,49,51,52	106.1957	30.500
100	100	0.25	0.225	11,15,22,25,29,30,31,33,37,39,40,42,43,49,52	106.1848	30.594
100	100	0.25	0.165	4,8,11,15,18,25,29,31,37,39,40,42,43,49,52	106.1809	30.547

Tabla 5.4 Mejores resultados al variar la probabilidad de mutación.

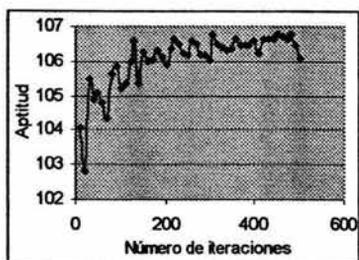
Las tablas con la información total recabada, se encuentran en el Apéndice 2. No obstante, con la intención de identificar el comportamiento que presentan los resultados al variar cada uno de los parámetros considerados, a continuación se presentan ocho gráficas que serán de gran ayuda. Las gráficas 5.1 y 5.2 corresponden a la tabla 5.1, las gráficas 5.3 y 5.4 corresponden a la tabla 5.2 y así sucesivamente. En cada uno de ellas se muestra la relación existente entre el parámetro variado, la aptitud alcanzada por el mejor de los cromosomas en la corrida y el tiempo requerido para completar la corrida.



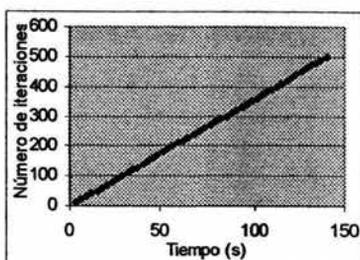
Gráfica 5.1



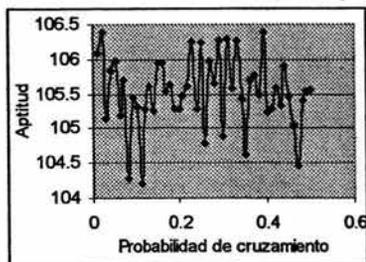
Gráfica 5.2



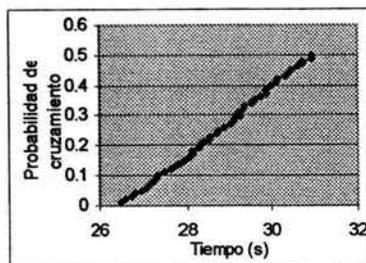
Gráfica 5.3



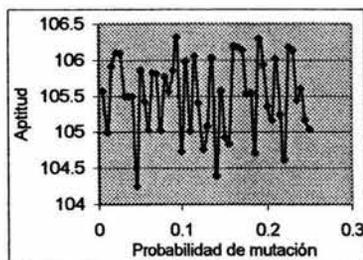
Gráfica 5.4



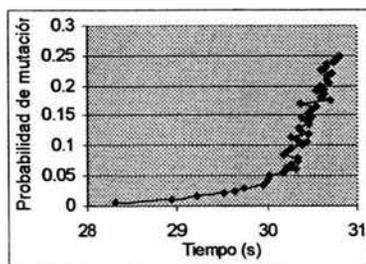
Gráfica 5.5



Gráfica 5.6



Gráfica 5.7



Gráfica 5.8

El primero de los parámetros analizados fue el número de cromosomas por considerar en cada una de las iteraciones. En la primera corrida, el valor asignado al parámetro fue de 10 y gradualmente se fue variando de 10 en 10 hasta ejecutar una corrida con 500 cromosomas.

A partir de la gráfica 5.1, se observa que el número de cromosomas considerado en cada iteración no fue un factor determinante para la obtención de soluciones buenas; ya que como se puede ver, existen corridas consecutivas que otorgaron resultados sumamente diferentes. Tampoco se puede afirmar que el incremento del número de cromosomas considerados, aumente la aptitud de las soluciones; como muestra de ello, se puede observar que más de la mitad de las corridas obtuvieron una solución con mejor aptitud que las últimas cuatro corridas ejecutadas, correspondientes a 470, 480, 490 y 500 cromosomas.

En lo que se refiere al tiempo requerido para ejecutar las corridas, se observa en la gráfica 5.2 una tendencia creciente superior a una lineal, ya que el tiempo usado por un par de corridas consecutivas iniciales varía aproximadamente por 3 segundos, mientras que el tiempo ocupado por un par de corridas consecutivas finales varía aproximadamente por 8 segundos.

Al variar el número de iteraciones por realizar entre 10 y 500 con saltos de 10 iteraciones, la tendencia de los resultados fue más clara que para los otros parámetros. De acuerdo a los resultados plasmados en la gráfica 5.3, el número de iteraciones por ejecutar sí es un factor determinante para encontrar soluciones buenas. A pesar de las variaciones en aptitud alcanzada entre corridas consecutivas, se observa una dependencia fuerte de la calidad de la solución con el número de iteraciones ejecutadas. Este hecho puede ser explicado, en parte, a partir del criterio elitista aplicado en el algoritmo, mismo que permite conservar a la mejor solución encontrada hasta el final de la corrida. Sin embargo, esta relación de dependencia tiende a ser menor mientras aumenta el número de iteraciones; como se ve en la gráfica 5.3, la diferencia entre la aptitud alcanzada por dos soluciones de dos corridas consecutivas, tiende a ser menor mientras aumenta el número de iteraciones, como pareciendo converger a la aptitud de la solución óptima del problema.

Como era de esperarse, la gráfica 5.4 muestra que la relación existente entre el tiempo transcurrido y el número de iteraciones por realizar en cada una de las corridas es lineal; se aprecian saltos aproximados de 3 segundos por cada incremento de 10 iteraciones.

Por su parte la probabilidad de cruzamiento fue variada entre 0.1 y 0.5 con saltos de 0.05 y la probabilidad de mutación entre 0.005 y 0.25 con saltos de 0.005. Ambas muestran ser, como en la teoría de los algoritmos genéticos, fuente de variabilidad de soluciones potenciales que en ocasiones permiten encontrar soluciones buenas y en otras

soluciones malas. Esto es precisamente lo que muestran las gráficas 5.5 y 5.7, donde se observa que no existe una tendencia definida que sugiera una relación de dependencia entre cada una de las dos probabilidades con la obtención de mejores soluciones.

Acerca del tiempo utilizado por las corridas al variar la probabilidad de cruzamiento, en la gráfica 5.6 se puede identificar una dependencia proporcional con variaciones ligeras originadas por las comparaciones entre la probabilidad de cruzamiento y los números aleatorios utilizados para definir si un cromosoma es seleccionado para aplicarle el operador cruzamiento.

En cuanto a la probabilidad de mutación y su relación con el tiempo, la gráfica 5.8 no muestra una tendencia tan clara, ya que para las primeras corridas consecutivas se observan incrementos de tiempo relativamente grandes, las corridas siguientes muestran que las diferencias entre los incrementos de tiempo disminuyen, pero más tarde se incrementan hasta que en la parte final de la serie se ven disminuidas nuevamente.

Ahora bien, analizando las mayores aptitudes encontradas entre los 200 resultados que se obtuvieron, se observa que las mayores aptitudes fueron alcanzadas al variar el número de iteraciones por ejecutar, seguidas de las encontradas al variar el número de cromosomas y la probabilidad de cruzamiento, y por último las encontradas al variar la probabilidad de mutación. Curiosamente la peor aptitud de las 200, corresponde a la solución encontrada en la segunda corrida ejecutada al variar el número de iteraciones por realizar.

Como resultado de este análisis sobre los parámetros genéticos del algoritmo genético 1, se podría sugerir definir un número de cromosomas entre 200 y 350; una probabilidad de cruzamiento entre 0.25 y 0.4; una probabilidad de mutación entre 0.09 y 0.23; y variar el número de iteraciones a partir de 300.

5.2.2 Efectos en el algoritmo genético 2

Para llevar a cabo el análisis sobre los parámetros genéticos del segundo algoritmo programado, que determina la ponderación que debe asignársele a cada uno de los activos, se utiliza nuevamente la matriz integrada por 54 activos en la que cada serie está compuesta por 220 rendimientos diarios; además se supone que la cartera será integrada por los activos 3, 7, 9, 16, 17, 27, 30, 34, 46 y 48.

Como se hiciera en el análisis de la sección previa, el algoritmo genético 2 se ejecutó 200 veces; 50 veces por cada parámetro propio del algoritmo genético. Los rangos considerados para variar los parámetros fueron: $10 \leq nc \leq 500$, $10 \leq ni \leq 500$, $0.01 \leq pc \leq 0.50$ y $0.005 \leq pm \leq 0.25$. Los mejores resultados se muestran a continuación.

nc	ni	pc	pm	Cromosoma	Aptitud	Tiempo (s)
230	100	0.25	0.01	0.1689,0.0816,0.0801,0.0685,0.0111, 0.2071,0.0556,0.1971,0.1300,0.0000	19.5509	135.593
420	100	0.25	0.01	0.1727,0.0758,0.0789,0.0651,0.0119, 0.2216,0.0506,0.2000,0.1234,0.0000	19.5494	304.172
410	100	0.25	0.01	0.1763,0.0737,0.0693,0.0688,0.0086, 0.2053,0.0620,0.1906,0.1454,0.0000	19.5493	293.937
320	100	0.25	0.01	0.1638,0.0704,0.0884,0.0665,0.0147, 0.2159,0.0555,0.2013,0.1235,0.0000	19.5476	209.046
470	100	0.25	0.01	0.1623,0.0787,0.0790,0.0591,0.0173, 0.2091,0.0610,0.1998,0.1333,0.0004	19.5463	358.313

Tabla 5.5 Mejores resultados al variar el número de cromosomas.

nc	ni	pc	pm	Cromosoma	Aptitud	Tiempo (s)
100	480	0.25	0.01	0.1708,0.0721,0.0736,0.0656,0.0161, 0.2139,0.0604,0.1927,0.1348,0.0000	19.5651	236.922
100	430	0.25	0.01	0.1725,0.0729,0.0796,0.0644,0.0130, 0.2117,0.0591,0.1939,0.1329,0.0000	19.5643	212.671
100	250	0.25	0.01	0.1758,0.0718,0.0701,0.0670,0.0134, 0.2118,0.0583,0.1986,0.1331,0.0001	19.5614	123.640
100	370	0.25	0.01	0.1671,0.0720,0.0729,0.0616,0.0274, 0.2124,0.0588,0.1941,0.1337,0.0000	19.5601	182.687
100	390	0.25	0.01	0.1696,0.0729,0.0711,0.0657,0.0264, 0.2055,0.0591,0.1939,0.1358,0.0000	19.5592	193.235

Tabla 5.6 Mejores resultados al variar el número de iteraciones.

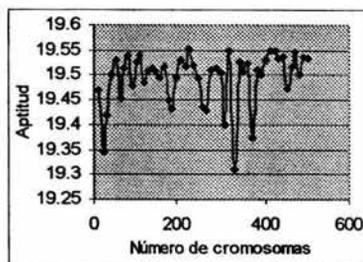
nc	ni	pc	pm	Cromosoma	Aptitud	Tiempo (s)
100	100	0.07	0.01	0.1727,0.0717,0.0806,0.0640,0.0150, 0.2132,0.0609,0.1956,0.1260,0.0003	19.5580	49.531
100	100	0.30	0.01	0.1720,0.0796,0.0841,0.0636,0.0176, 0.1974,0.0613,0.1893,0.1351,0.0000	19.5504	49.719
100	100	0.24	0.01	0.1648,0.0723,0.0885,0.0688,0.0161, 0.2041,0.0573,0.1890,0.1381,0.0010	19.5399	49.546
100	100	0.16	0.01	0.1800,0.0717,0.0569,0.0604,0.0219, 0.2169,0.0553,0.1910,0.1458,0.0001	19.5383	49.500
100	100	0.18	0.01	0.1748,0.0723,0.0752,0.0710,0.0075, 0.2052,0.0659,0.1958,0.1313,0.0010	19.5376	49.516

Tabla 5.7 Mejores resultados al variar la probabilidad de cruzamiento.

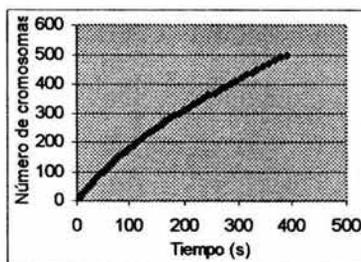
nc	ni	pc	pm	Cromosoma	Aptitud	Tiempo (s)
100	100	0.25	0.020	0.1716,0.0652,0.1073,0.0754,0.0127, 0.1968,0.0558,0.1836,0.1316,0.0000	19.5071	49.563
100	100	0.25	0.025	0.1775,0.0731,0.0682,0.0752,0.0082, 0.2354,0.0534,0.1963,0.1108,0.0019	19.5022	49.547
100	100	0.25	0.010	0.1608,0.0564,0.0791,0.0495,0.0341, 0.2170,0.0564,0.1958,0.1509,0.0000	19.5014	49.469
100	100	0.25	0.030	0.1788,0.0653,0.0779,0.0808,0.0248, 0.2019,0.0625,0.1757,0.1323,0.0000	19.5007	49.609
100	100	0.25	0.045	0.1763,0.0743,0.1048,0.0637,0.0028, 0.1957,0.0701,0.1902,0.1216,0.0005	19.4953	49.672

Tabla 5.8 Mejores resultados al variar la probabilidad de mutación.

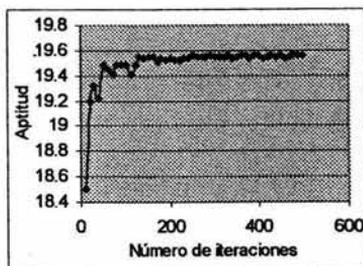
Las tablas con la información total obtenida, se encuentran en el Apéndice 2. Las gráficas que se muestran a continuación (gráficas 5.9 – 5.16), plasman la información que se consiguió a lo largo de las 200 ejecuciones que se hicieron del algoritmo genético 2 que determina las ponderaciones de cada activo que integra la cartera de inversión.



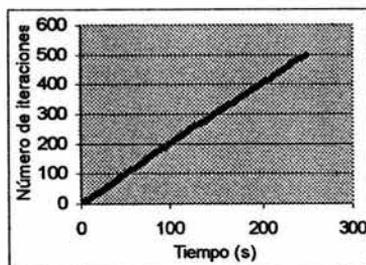
Gráfica 5.9



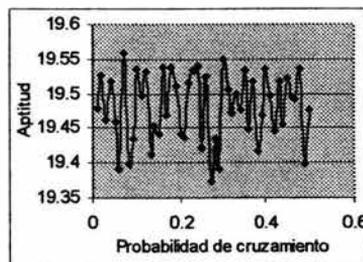
Gráfica 5.10



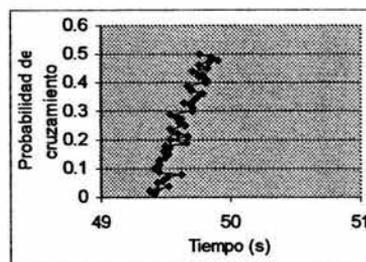
Gráfica 5.11



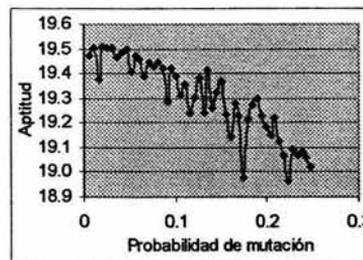
Gráfica 5.12



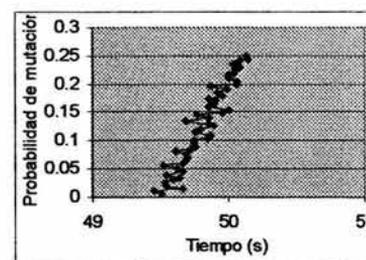
Gráfica 5.13



Gráfica 5.14



Gráfica 5.15



Gráfica 5.16

A partir del análisis de las gráficas mostradas arriba, se pueden establecer criterios para definir los parámetros genéticos del algoritmo genético 2.

El primer parámetro analizado fue el número de cromosomas por considerar en cada iteración. De acuerdo a las gráficas 5.9 y 5.10, se puede establecer que no existe relación entre la aptitud alcanzada por la mejor solución del algoritmo genético y el número

de cromosomas considerados en cada iteración; como en el caso del algoritmo genético 1, se observa que existen corridas con un número grande de cromosomas considerados, pero que encontraron una solución con menor aptitud que corridas ejecutadas considerando una menor cantidad de cromosomas. Además, de acuerdo a los resultados obtenidos, se puede decir que no es posible identificar rangos de los que sea más conveniente tomar el valor del parámetro de número de cromosomas, porque en la gráfica 5.9 se observan diferencias relativamente grandes entre corridas consecutivas; el ejemplo más claro de esta situación corresponde a las corridas ejecutadas considerando 310, 320, 330 y 340 cromosomas.

En lo que se refiere a la relación entre el tiempo utilizado por la corrida y el número de cromosomas considerado, se puede identificar como una relación ligeramente superior a una relación lineal.

El análisis del parámetro de número de iteraciones por realizar en cada corrida, se basa en las gráficas 5.11 y 5.12; estas gráficas muestra tendencias muy claras de su comportamiento. La tendencia mostrada en la primera de ellas, sugiere una relación fuerte de dependencia entre la aptitud alcanzada por la mejor solución encontrada y el número de iteraciones ejecutadas. Además se pudo observar que, a partir de las 250 iteraciones, las aptitudes alcanzadas por los mejores cromosomas encontrados varían ligeramente; caso contrario se observa en las corridas con un número de iteraciones entre 10 y 110, donde las diferencias entre las aptitudes encontradas son más grandes.

La relación existente entre el tiempo necesario para completar una corrida y el número de iteraciones por ejecutar, de acuerdo a los resultados obtenidos, es lineal; esta relación se deduce con base en los incrementos aproximados de 5 segundos que se presentaron por cada 10 iteraciones más que se realizaban.

El tercero de los parámetros en ser variado fue la probabilidad de cruzamiento. Los resultados obtenidos y la gráfica 5.13, sugieren que el operador cruzamiento sólo facilita el intercambio de información entre las soluciones potenciales, pero no dan lugar a pensar en una relación que indique si el incremento o la disminución del valor adoptado por este parámetro afecta la aptitud alcanzada por la mejor solución encontrada. Por su parte, la gráfica 5.14, muestra que un incremento en el valor asignado a la probabilidad de cruzamiento, causa un incremento ligero en el tiempo requerido para completar la corrida.

En cuanto a los resultados obtenidos al variar la probabilidad de mutación, correspondientes a las gráficas 5.15 y 5.16, se pueden interpretar como una muestra clara de que el incremento de la probabilidad de mutación, no favorece la obtención de mejores soluciones e incrementa el tiempo necesario para completar la corrida en un grado mayor que una variación sobre la probabilidad de cruzamiento.

A partir de las observaciones señaladas en los párrafos anteriores, se establece que en el algoritmo genético 2, el parámetro que tiene una mayor influencia positiva en la obtención de una solución buena al problema, es el número de iteraciones consideradas. Por ello se propone que este parámetro sea el que más se varíe, incrementándolo a partir de 250 iteraciones. Por el contrario, la probabilidad de mutación es el parámetro que afecta más la obtención de soluciones buenas; por ello, se recomienda utilizar una probabilidad de mutación considerablemente baja. Por su parte el número de cromosomas y la probabilidad de cruzamiento, no mostraron un gran efecto sobre las soluciones buenas al momento de ser variados, por lo que es conveniente fijarlos en un determinado valor, por ejemplo, en los valores que otorgaron las mejores aptitudes dentro de sus respectivas corridas.

Este análisis es aplicable al tercer y cuarto algoritmo, porque la codificación de soluciones es igual, y los parámetros y las funciones de aptitud son muy parecidos.

5.2.3 Efectos en el algoritmo genético 5

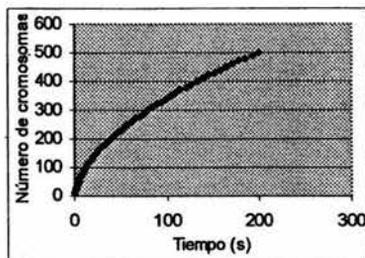
Como se explicó en el capítulo anterior, el algoritmo genético 5 es un algoritmo creado a partir del algoritmo genético 2, pero encierra un menor grado de complejidad. Esto es debido a que sus parámetros de entrada financieros son mucho más simples, sólo se trata de cuatro valores, dejando a un lado la matriz de rendimientos y el vector de los activos por combinar en la cartera de inversión; además, cada solución potencial probable consta únicamente de un número entre 0 y 1.

Para analizar el algoritmo genético 5, se le ejecutó 200 veces variando los parámetros genéticos como se ha hecho antes. Como parámetros financieros se supuso un rendimiento diario de la cartera de inversión de 0.0031 con riesgo de 0.0071, un rendimiento diario del activo libre de riesgo de 0.0002 y un grado de aversión al riesgo del inversionista de 100.

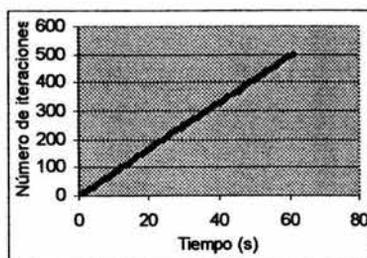
Los resultados obtenidos son claros, mismos que se muestran en las tablas ubicadas en el Apéndice 2. Se afirma que los resultados son claros debido a que sólo en 19 de las 200 ejecuciones, el algoritmo genético 5 no encontró la solución óptima. Por esta razón resulta innecesario mostrar, como se había hecho antes, las tablas con los cinco mejores resultados y las gráficas que confrontaban el parámetro analizado y la aptitud alcanzada en cada ejecución.

Lo que sí es importante resaltar, son los efectos causados por la variación de cada uno de los parámetros en el tiempo de ejecución requerido para completar la corrida. Además, es necesario dar una interpretación más específica sobre los efectos de las variaciones sobre los parámetros de carácter genético del quinto algoritmo.

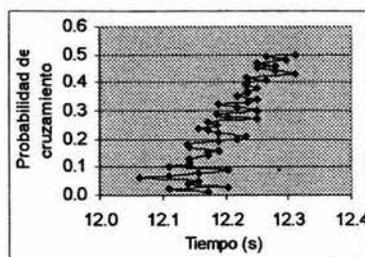
A continuación se muestran las gráficas 5.17 a 5.20 que muestran la relación entre la variación de cada parámetro y el tiempo utilizado por la correspondiente corrida.



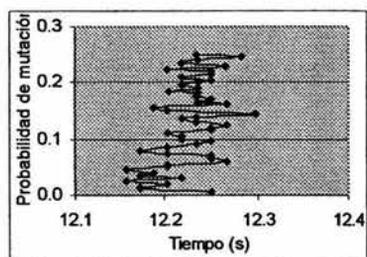
Gráfica 5.17



Gráfica 5.18



Gráfica 5.19



Gráfica 5.20

Sobre el parámetro correspondiente al número de cromosomas considerados, se observó que sólo tres de las primeras cuatro ejecuciones no lograron encontrar la solución óptima, dichas corridas corresponden a los valores de 10, 30 y 40 cromosomas. Fuera de dichas ejecuciones, el algoritmo genético 5 mostró la misma solución una y otra vez.

De acuerdo a los resultados que se muestran en la gráfica 5.17, se observa que los tiempos utilizados por las corridas van incrementándose; entre las primeras corridas consecutivas, las diferencias de tiempo utilizado fueron aproximadamente de 1 segundo y se incrementaron hasta llegar a casi 8 segundos de diferencia que registraron las últimas dos corridas.

Cuando se varió el parámetro que indica el número de iteraciones por ejecutar, se observó que sólo las corridas con 10 y 110 iteraciones no otorgaron el resultado óptimo.

La relación entre el tiempo transcurrido en la corrida y el número de iteraciones por realizar, según los datos de la gráfica 5.18, es lineal. La diferencia de tiempo transcurrido entre corridas consecutivas varía entre 1 y 2 segundos.

En lo que respecta a la probabilidad de cruzamiento, de las 50 corridas en las que se varió este parámetro, 5 mostraron una solución distinta a la óptima. En dichas corridas, los valores asignados a este parámetro fueron: 0.06, 0.10, 0.21, 0.26 y 0.36. Esto permite pensar nuevamente, que el parámetro de cruzamiento no es un factor sumamente determinante para que el algoritmo genético encuentre soluciones buenas; por ello se observa que los parámetros que no encontraron la solución óptima, se encuentra distribuidos a lo largo de todo el rango que se consideró ($0.01 \leq pc \leq 0.5$), tal vez con una ligera tendencia hacia la primera sección del rango referido. Sobre los tiempos registrados, se puede decir que la variación de la probabilidad de cruzamiento, no afecta visiblemente el tiempo necesario para culminar la ejecución del algoritmo.

Como en el algoritmo genético 2, se encontró que los valores mayores asignados a la probabilidad de mutación, impiden al algoritmo encontrar soluciones buenas tan frecuentemente como lo hace con probabilidades bajas. Prueba de ello es el hecho reflejado por las nueve corridas que no lograron encontrar la mejor solución en los que la probabilidad de mutación asignada fue de: 0.03, 0.17, 0.185, 0.195, 0.215, 0.235, 0.24, 0.245 y 0.25. Dado que el rango utilizado para variar la probabilidad de mutación fue $0.005 \leq pm \leq 0.25$, se observa que la mayoría de las ejecuciones que no encontraron la mejor solución, adquirieron el valor del parámetro de la parte final del rango considerado.

Para este algoritmo se sugiere usar un número fijo mayor a 100 de cromosomas y de iteraciones, así como otorgar probabilidades bajas al cruzamiento y la mutación.

5.3 Creación de una cartera de inversión con un universo de activos reducido

En esta sección se crea una cartera de inversión, suponiendo que el universo de activos disponibles para el inversionista consta únicamente de 7 activos. A partir los siete activos disponibles, el inversionista desea formar una cartera de inversión que incluya a tres de ellos.

Los activos disponibles para el inversionista son: BACHOCOUBL (Industrias Bachoco, S.A. de C.V.), CEL* (Grupo Iusacell, S.A. de C.V.), DESCB (DESC, S.A. de C.V.), GEOB (Corporación GEO, S.A. de C.V.), PE&OLES* (Industrias Peñoles, S.A. de C.V.), SAVIAA (SAVIA, S.A. de C.V) y USCOMB-1 (US Commercial Corporation, S.A. de C.V.). Según el orden en el que los activos han sido mencionados, se identificará a cada uno de ellos a lo largo de esta sección; es decir, BACHOCOUBL es el activo 1, CEL* es el activo 2 y así sucesivamente.

La matriz de rendimientos de los activos utilizada en este ejemplo, es una matriz de 220×7 . En otras palabras, se cuenta con 220 rendimientos diarios de cada uno de los 7 activos disponibles para el inversionista. Las series de rendimientos corresponden a los registrados entre el 4 de agosto de 2003 y el 4 de junio de 2004.

5.3.1 Selección de activos

Como se mencionó antes, se desea seleccionar 3 de los 7 activos para conformar la cartera de inversión. Para ello, se ejecutó el algoritmo genético 1 en 5 ocasiones considerando 220 cromosomas, 310 iteraciones, probabilidad de cruzamiento de 0.39 y probabilidad de mutación de 0.095.

En las 5 corridas ejecutadas, la solución encontrada fue 1,5,7 con una aptitud de 3.4371; en otras palabras, de las soluciones potenciales probables evaluadas por el algoritmo genético en las ejecuciones, la solución 1,5,7 fue la que obtuvo mayor aptitud.

En este caso, el conjunto de soluciones potenciales probables es un conjunto finito de cardinalidad 35, por lo que es posible calcular cada una de las aptitudes y comprobar que la secuencia 1,5,7 realmente alcanza la mejor aptitud.

Para calcular la aptitud de cada una de las soluciones potenciales probables, es necesario conocer el rendimiento promedio, la desviación estándar y los coeficientes de correlación por cada activo disponible. Estos datos se muestran a continuación.

Activos	Riesgo	Rendimiento promedio
1. BACHOCOUBL	0.01698159781906	0.00011508094538
2. CEL*	0.18754817191963	0.01478988074872
3. DESCB	0.02532861040418	-0.00104175393532
4. GEOB	0.10926266530352	-0.00425739076597
5. PE&OLES*	0.03418495820655	0.00313094899528
6. SAVIAA	0.12590932119419	-0.00378433036468
7. USCOMB-1	0.01699385667735	0.00113134567008

Tabla 5.9 Rendimiento y riesgo por activo

	1	2	3	4	5	6	7
1	1.00000	0.03133	0.07273	0.01983	-0.04074	-0.06430	0.01886
2	0.03133	1.00000	0.15228	-0.00768	-0.02390	0.01398	0.00744
3	0.07273	0.15228	1.00000	-0.04977	0.12506	0.06955	0.04548
4	0.01983	-0.00768	-0.04977	1.00000	0.23632	-0.02701	0.02645
5	-0.04074	-0.02390	0.12506	0.23632	1.00000	0.02168	-0.04711
6	-0.06430	0.01398	0.06955	-0.02701	0.02168	1.00000	0.00587
7	0.01886	0.00744	0.04548	0.02645	-0.04711	0.00587	1.00000

Matriz de coeficientes de correlación

A partir de los datos anteriores, el algoritmo genético 1 evaluó las soluciones potenciales probables de acuerdo a la función de aptitud. Como se mencionó antes, considerando únicamente 7 activos, las soluciones potenciales probables son el conjunto de combinaciones de 7 en 3 elementos; es decir, existen 35 soluciones potenciales probables.

La función aptitud que se utilizó en el algoritmo genético 1 es:

$$F = |cr| + cd + \frac{(na-1)na}{2} + c + \sum_{i=1}^{na} [r_i - d_i - \sum_{j=i+1}^{na} C_{ij}] \quad (5.1)$$

Así, por ejemplo, el cálculo de la aptitud de la mejor solución sería:

$$F(1,5,7) = |-0.0091| + 0.4227 + \frac{(3-1) \cdot 3}{2} + 0.0001 + 0.0052 = 3.4371$$

Para cada solución potencial probable se puede realizar el cálculo de su aptitud análogamente a como se ha calculado la aptitud de la secuencia 1,5,7. Las aptitudes de las 35 soluciones potenciales probables son las siguientes:

Solución	Aptitud								
1,2,3	2.9596	1,3,6	3.1810	1,6,7	3.3090	2,4,7	3.1036	3,5,6	3.0285
1,2,4	3.0853	1,3,7	3.2357	2,3,4	3.0244	2,5,6	3.0866	3,5,7	3.2352
1,2,5	3.2445	1,4,5	3.0551	2,3,5	2.9483	2,5,7	3.2758	3,6,7	3.1391
1,2,6	3.1316	1,4,6	3.2433	2,3,6	2.8673	2,6,7	3.0863	4,5,6	2.9266
1,2,7	3.1688	1,4,7	3.2205	2,3,7	3.0117	3,4,5	3.1995	4,5,7	3.0558
1,3,4	3.2323	1,5,6	3.3377	2,4,5	2.9098	3,4,6	3.1695	4,6,7	3.1675
1,3,5	3.2006	1,5,7	3.4371	2,4,6	3.0366	3,4,7	3.2540	5,6,7	3.2748

Tabla 5.10 Soluciones potenciales y aptitudes.

En la tabla 5.10 se observa que la mayor aptitud es justamente la alcanzada por la solución 1,5,7. De esta forma, la cartera de inversión estará formada por los activos BACHOCOUBL, PE&OLES* y USCOMB-1.

5.3.2 Determinación de ponderaciones

Una vez que han sido seleccionados los tres activos financieros que integrarán la cartera de inversión, se desea obtener los pesos que deben ser asignados a cada uno de los activos para crear una cartera de inversión que, según información obtenida a partir de la base datos, presente el mejor cociente rendimiento - varianza; como se explicó en su momento, ese cociente refleja directamente los criterios de varianza mínima y del inversionista insaciable.

Para determinar las ponderaciones de cada activo que integrará la cartera de inversión, el algoritmo genético 2 se ejecutó 10 veces. En este caso los parámetros financieros son la matriz de rendimientos de 220×7 , misma que fue utilizada en el algoritmo genético previo, y el vector (1,5,7) que indica los activos seleccionados.

Sobre los parámetros genéticos, de acuerdo a las recomendaciones establecidas en el análisis previo, se utilizaron 230 cromosomas durante 480 iteraciones, mientras que las probabilidades de cruzamiento y mutación se fijaron en 0.07 y 0.02 respectivamente.

Antes de mostrar los resultados que se obtuvieron en las 10, debe hacerse una aclaración del por qué los resultados serán mostrados en formato largo. Como se explicó en el capítulo 4, la forma en que algoritmos genéticos 2, 3, 4 y 5 fueron programados es considerando que cada variable de una solución, mostrará una exactitud de 4 cifras decimales. Sin embargo, existen casos en los que el redondeo sobre las cuatro cifras decimales de cada variable de una solución provoca que la suma de todas las variables sea distinta de uno; en otras palabras, algunas veces el redondeo provoca que una solución originalmente factible (considerando un formato largo) se convierta en una solución no factible (considerando un formato corto). También es pertinente mencionar que este problema no es exclusivo de los algoritmos aquí programados; éste es un problema que se presenta en la práctica del cálculo numérico, ya que las soluciones calculadas por una computadora no son soluciones matemáticas exactas.

Los resultados de las 10 corridas ejecutadas del algoritmo genético 2 fueron:

Corrida	Tiempo (s)	Cromosoma	Aptitud
1	313.3910	0.28395719077791,0.22906128534157,0.48698152388052	8.94769389776655
2	313.6560	0.28395920259620,0.22906815020862,0.48697264719518	8.94769390327434
3	313.1560	0.28394077448747,0.22908124525437,0.48697798025816	8.94769387949236
4	312.5790	0.28396281657151,0.22906602254428,0.48697116088421	8.94769390355493
5	314.1560	0.28396065675341,0.22905049566295,0.48698884758364	8.94769387545605
6	314.8590	0.28397196686994,0.22905858438761,0.48696944874245	8.94769389652128
7	313.1560	0.28402366863905,0.22905252392432,0.48692380743663	8.94769378708561
8	312.6870	0.28398838942496,0.22907350749196,0.48693810308308	8.94769387253771
9	314.0620	0.28400888395336,0.22920599666852,0.48678511937812	8.94769181935352
10	315.6250	0.28395002860957,0.22906732786573,0.48698264352470	8.94769389776413

Tabla 5.11 Resultados de las corridas del algoritmo genético 2.

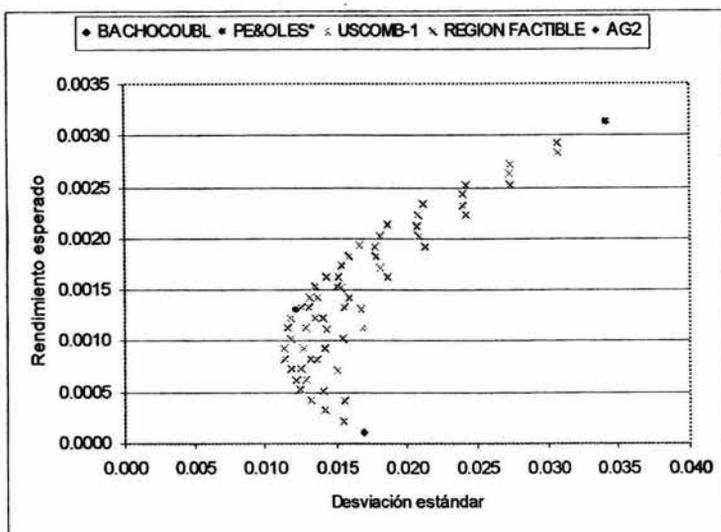
Como se puede observar en la tabla 5.11, las ponderaciones encontradas y las aptitudes alcanzadas por cada solución son casi idénticas. Basándose en la columna de las aptitudes se observa que la mejor solución fue la encontrada en la corrida 4, pero obsérvese lo que sucede con dicha solución. Si se suman las tres entradas del mejor cromosoma evaluado en la corrida cuatro, utilizando el formato largo, se observa que la solución es factible; sin embargo, al sumar las entradas después de redondear cada una de ellas a cuatro cifras ($0.2840 + 0.2291 + 0.4870 = 1.0001$) la solución pierde la factibilidad.

En tal situación, se decidió asumir como mejor solución aquella que mayor aptitud presentará y que conservara la factibilidad inclusive después de aplicarle redondeo; en este caso, tal solución corresponde a la encontrada en la corrida 3 que tiene aptitud ligeramente menor a la encontrada en la corrida 4.

De esta forma, la mejor solución encontrada por el algoritmo genético 2 fue (0.2839, 0.2291, 0.4870). Esto quiere decir que la cartera de inversión conformada en un 28.39% por el activo BACHCOCOUBL, en un 22.91% por el activo PE&OLES* y en un 48.70% por el activo USCOMB-1, mostró el mejor cociente rendimiento – varianza entre las carteras que fueron evaluadas por el algoritmo genético 2; a dicha cartera se le denotará como AG2.

Con base en los datos mostrados en la tabla 5.9, es posible calcular el rendimiento y el riesgo que posee la cartera de inversión AG2. Después de realizar los cálculos correspondientes, justo como se explicó en el capítulo 2, se encontró que la cartera de inversión AG2 tiene un rendimiento esperado diario de 0.1301% y un riesgo de 0.012058.

En la siguiente gráfica se muestran algunos puntos de la región factible formada por los tres activos considerados y también la ubicación de la cartera de inversión AG2.



Gráfica 5.21 Región factible y AG2

A la cartera AG2 se le puede describir como la cartera eficiente que muestra el mayor cociente rendimiento – varianza.

5.3.3 Punto de varianza mínima

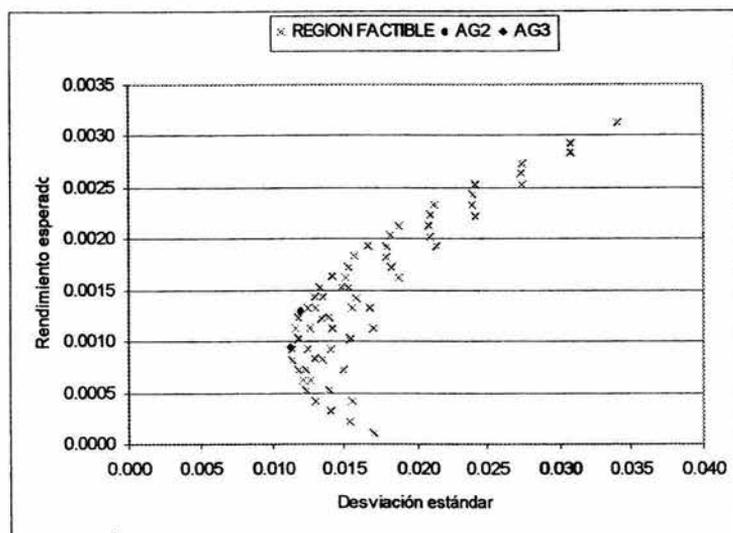
Mediante el algoritmo genético 3 se determinaron los pesos que definen la cartera de riesgo mínimo. El algoritmo fue ejecutado en 10 ocasiones con 230 cromosomas, 480 iteraciones, probabilidad de cruzamiento de 0.07 y probabilidad de mutación de 0.02.

Corrida	Tiempo (s)	Cromosoma	Aptitud
1	309.4690	0.43642282093760,0.12648588949923,0.43709128956317	0.03415972410271
2	309.7340	0.43634088429699,0.12629147147533,0.43736764422768	0.03415972419911
3	309.8130	0.43636928991407,0.12624378109453,0.43738692899140	0.03415972420281
4	313.8120	0.43638690412712,0.12623084853009,0.43738224734279	0.03415972420269
5	309.9850	0.43547758284600,0.12702463228779,0.43749778486621	0.03415972321970
6	310.1090	0.43639575971731,0.12624227031802,0.43736196996467	0.03415972420252
7	310.1870	0.43639970245475,0.12623977188197,0.43736052566328	0.03415972420245
8	309.9060	0.43635911641452,0.12728176717096,0.43635911641452	0.03415972256868
9	314.0000	0.43638126491647,0.12623060859188,0.43738812649165	0.03415972420273
10	310.0310	0.43636130356685,0.12625096227868,0.43738773415447	0.03415972420263

Tabla 5.12 Resultados de las corridas del algoritmo genético 3.

En la tabla 5.12 se muestran los resultados obtenidos. La mejor solución encontrada (corrida 3), indica que la cartera de inversión de varianza mínima está formada en un 43.64% por BACHOCOUBL, en un 12.62% por PE&OLES* y en un 43.74% por USCOMB-1; a esta nueva cartera se le llamará AG3.

En la siguiente gráfica se puede muestra la ubicación de las dos carteras encontradas hasta el momento en la región factible.



Gráfica 5.21 Región factible, AG2 y AG3

La cartera AG3, correspondiente al punto de varianza mínima, posee un rendimiento esperado diario de 0.0940% y un riesgo de 0.011191.

5.3.4 Inclusión del activo libre de riesgo

Para encontrar la cartera de inversión mencionada en el teorema de un fondo, se ejecutó el algoritmo genético 4 considerando que el rendimiento diario del activo libre de riesgo era de 0.000182. Los parámetros considerados para las 10 corridas ejecutadas, son los mismos que han sido utilizados en las ejecuciones de los algoritmos genéticos 2 y 3.

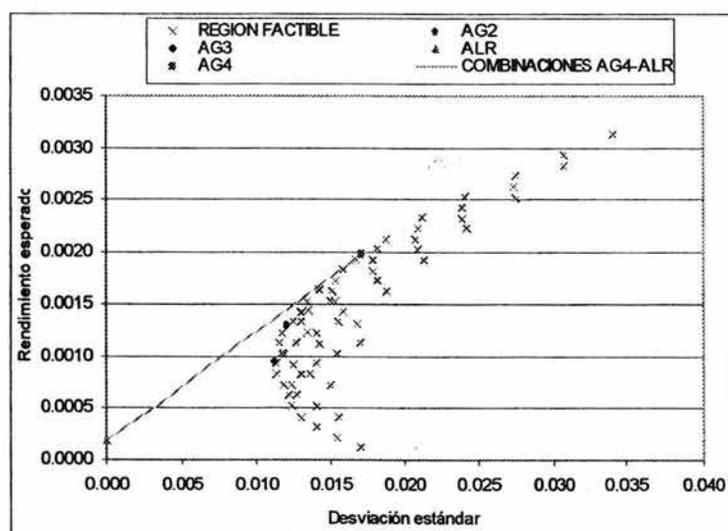
Los resultados obtenidos se muestran en la tabla 5.13.

Corrida	Tiempo (s)	Cromosoma	Aptitud
1	315.5320	0,0.42443654544744,0.57556345455256	0.10507559021197
2	314.1250	0,0.42444085381410,0.57555914618590	0.10507559020070
3	314.6720	0,0.42442484662577,0.57557515337423	0.10507559020393
4	314.4070	0,0.42421970563864,0.57578029436136	0.10507558088033
5	315.4060	0,0.42443198159333,0.57556801840667	0.10507559021555
6	314.3440	0,0.42443566370643,0.57556433629357	0.10507559021333
7	315.6100	0,0.42443593018306,0.57556406981694	0.10507559021295
8	315.0940	0,0.42443668676611,0.57556331323389	0.10507559021172
9	314.0630	0,0.42444012056007,0.57555987943993	0.10507559020316
10	313.0620	0,0.42443135896548,0.57556864103452	0.10507559021538

Tabla 5.13 Resultados de las corridas del algoritmo genético 4

La mejor aptitud registrada, fue otorgada en la corrida 5. A partir de ese resultado, se supo que la cartera de inversión formada en un 0.00% por BACHOCOUBL, en un 42.44% por PE&OLES* y en un 57.56% por USCOMB-1 es la cartera a la que se refiere el Teorema de un Fondo de Inversión; a esta nueva cartera se le llamará AG4.

En la grafica siguiente se muestra la ubicación de la cartera AG4.



Gráfica 5.22 Región factible, AG2, AG3 y AG4

En la gráfica 5.22, además de observar la ubicación de la cartera AG4, se muestra la recta que representa a las posibles combinaciones entre el activo libre de riesgo (ALR) y la cartera de inversión AG4.

La cartera de inversión AG4, correspondiente a la cartera que hace válido el Teorema de un Fondo de Inversión, posee un rendimiento esperado diario de 0.1980% y un riesgo de 0.017111.

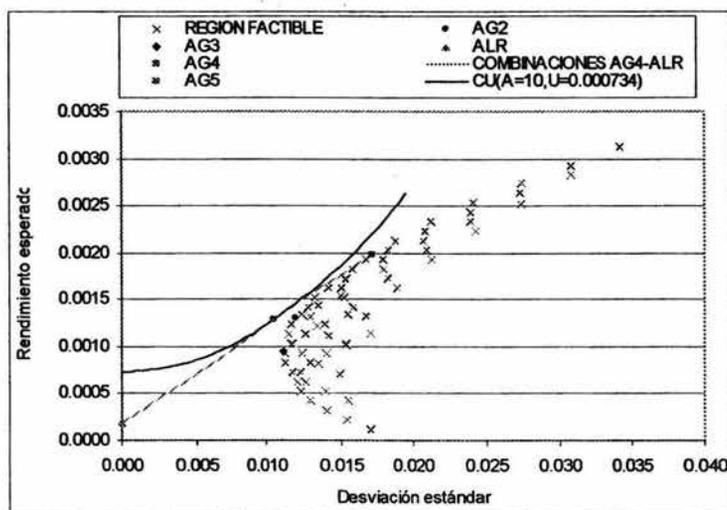
5.3.5 Inclusión del grado de aversión al riesgo del inversionista

Como último paso en el proceso de creación de la cartera de inversión, se considera el grado de aversión al riesgo del inversionista y se ubica su cartera óptima. Para ello se consideró que el inversionista tiene un grado de aversión al riesgo $A = 10$.

Se ejecutaron 5 corridas del algoritmo genético 5; los parámetros genéticos utilizados fueron 230 cromosomas, 480 iteraciones, probabilidad de cruzamiento de 0.01 y probabilidad de mutación de 0.005. Los parámetros financieros utilizados fueron el rendimiento diario de la cartera de inversión AG4, 0.001980; su desviación estándar, 0.017111; el rendimiento diario del activo libre de riesgo, 0.000182; y el grado de aversión al riesgo del inversionista, $A=10$.

De los 5 resultados obtenidos en las ejecuciones del algoritmo genético 5, cuatro ofrecieron como mejor solución el valor $y = 0.6141$ cuya aptitud alcanzada fue de 0.000734. Este valor y indica que la cartera de inversión formada en un 61.41% por la cartera AG4 y en un 38.59% por el activo libre de riesgo, es la cartera de inversión óptima para el inversionista con grado de aversión al riesgo $A=10$. A esta nueva cartera se le llamará AG5, cuyo rendimiento diario esperado es de 0.1286% y su riesgo de 0.010508.

En la gráfica 5.23 se pueden observar cada uno de los resultados obtenidos por los algoritmos genéticos.



Gráfica 5.23 Región factible, AG2, AG3, AG4 y AG5

En la gráfica 5.23 se pueden observar las cuatro carteras de inversión encontradas a partir de los algoritmos genéticos programados. Además se observa una de las curvas de utilidad de un inversionista con grado de aversión al riesgo $A=10$; siendo más precisos, esa curva representa la curva tangente a la recta generada por la cartera de inversión AG4 y el activo libre de riesgo, que corresponde a una utilidad de 0.000734.

Así pues, se consideró un inversionista que dispone de un universo de activos riesgosos (integrado por BACHOCOUBL, CEL*, DESCB, GEOB, PE&OLES*, SAVIAA y USCOMB-1); deseando seleccionar tres de esos activos para agregarlos a un activo libre de riesgo en su cartera de inversión; y que presenta un grado de aversión al riesgo $A=10$. Mediante los algoritmos genéticos, principalmente 1, 4 y 5, el inversionista encontró que su cartera óptima es la que está constituida en un 0.00% por BACHOCOUBL, en un 26.06% por PE&OLES*, en un 35.35% por USCOMB-1 y en un 38.59% por el activo libre de riesgo.

5.4 Creación y administración de una cartera de inversión

En esta sección se creará y administrará una cartera de inversión considerando las características siguientes. Existe un inversionista que dispone de un universo de activos riesgosos integrado por 54 activos (acciones cotizantes en la Bolsa Mexicana de Valores); él desea seleccionar 10 activos para conformar una cartera de mercado, incluir un activo libre de riesgo (Cetes a 28 días) y considerar que posee un grado de aversión al riesgo $A = 100$ para encontrar su cartera óptima. Además, desea administrar esta primera cartera durante dos semanas, para después recalcularla con la información que en esas dos semanas se genere y así administrar su nueva cartera dos semanas más.

5.4.1 Creación de la cartera de inversión

El inversionista cuenta con los rendimientos de 54 acciones que se registraron durante el 4 de agosto de 2003 y el 4 de junio de 2004 en la Bolsa Mexicana de Valores. De esta forma, su base de datos es la matriz M de 220×54 .

A continuación se muestran las 54 acciones consideradas originalmente y el número de columna que ocupa cada activo en la matriz M .

ACTIVO	CLAVE	EMPRESA
1	ALFAA	ALFA, S.A. DE C.V.
2	AMTELA1	AMERICA TELECOM, S.A. DE C.V.
3	AMXL	AMERICA MOVIL, S.A. DE C.V.
4	ARA*	CONSORCIO ARA, S.A. DE C.V.
5	ASURB	GRUPO AEROPORTUARIO DEL SURESTE, S.A. DE C.V.
6	BACHOCOUBL	INDUSTRIAS BACHOCO, S.A. DE C.V.
7	BIMBOA	GRUPO BIMBO, S.A. DE C.V.
8	CEL*	GRUPO IUSACELL, S. A. DE C. V.
9	CEMEXCPO	CEMEX, S.A. DE C.V.
10	CIEB	CORP. INTERAMERICANA DE ENTRETO., S.A. DE C.V.
11	CMRB	CORP. MEXICANA DE RESTAURANTES, S.A. DE C.V.
12	COMERCIUBC	CONTROLADORA COMERCIAL MEXICANA, S.A. DE C.V.
13	CONTAL*	GRUPO CONTINENTAL, S.A.
14	DESCB	DESC, S.A. DE C.V.
15	EKCO*	EKCO, S.A.

16	ELEKTRA*	GRUPO ELEKTRA, S.A. DE C.V.
17	FEMSAUBD	FOMENTO ECONÓMICO MEXICANO, S.A. DE C.V.
18	FRAGUAB	CORPORATIVO FRAGUA, S.A. DE C.V.
19	GCARSOA1	GRUPO CARSO, S.A. DE C.V.
20	GCC*	GRUPO CEMENTOS DE CHIHUAHUA, S.A. DE C.V.
21	GEOB	CORPORACION GEO, S.A. DE C.V.
22	GFINBURO	GRUPO FINANCIERO INBURSA, S.A. DE C.V.
23	GFNORTEO	GRUPO FINANCIERO BANORTE, S.A. DE C.V.
24	GIGANTE*	GRUPO GIGANTE, S.A. DE C.V.
25	GISSA*	GRUPO INDUSTRIAL SALTILLO, S. A. DE C.V.
26	GMEXICOB	GRUPO MEXICO, S.A. DE C.V.
27	GMODELOC	GRUPO MODELO, S.A. DE C.V.
28	GRUMAB	GRUMA, S.A. DE C.V.
29	HILASALA	HILASAL MEXICANA S.A. DE C.V.
30	HOGARB	CONSORCIO HOGAR, S.A. DE C.V.
31	HYLSAMXB	HYLSAMEX, S.A. DE C.V.
32	ICA*	EMPRESAS ICA SOCIEDAD CONTROLADORA, S.A. DE C.V.
33	ICHB	INDUSTRIAS CH, S.A. DE C.V.
34	IMSAUBC	GRUPO IMSA, S.A. DE C.V.
35	KIMBERA	KIMBERLY - CLARK DE MEXICO S.A DE C.V.
36	KOFL	COCA-COLA FEMSA, S.A. DE C.V.
37	LIVEPOLC-1	EL PUERTO DE LIVERPOOL, S.A. DE C.V.
38	MASECAB	GRUPO INDUSTRIAL MASECA, S.A. DE C.V.
39	MOVILAB	BIPER, S.A. DE C.V.
40	NADROB	NADRO, S.A. DE C.V.
41	PE&OLES*	INDUSTRIAS PEÑOLES, S. A. DE C. V.
42	POSADASL	GRUPO POSADAS, S.A. DE C.V.
43	SAVIAA	SAVIA, S.A. DE C.V.
44	SIMECB	GRUPO SIMEC, S.A. DE C.V.
45	SORIANAB	ORGANIZACION SORIANA, S.A. DE C.V.
46	TELECOMA1	CARSO GLOBAL TELECOM, S.A. DE C.V.
47	TELMEXL	TELEFONOS DE MEXICO, S.A. DE C.V.
48	TLEVISACPO	GRUPO TELEVISA, S.A.
49	TMMA	GRUPO TMM, S.A.
50	TVAZTCACPO	TV AZTECA, S.A. DE C.V.
51	USCOMB-1	US COMMERCIAL CORP, S.A. DE C.V.
52	VALLEB	JUGOS DEL VALLE, S.A. DE C.V.
53	VITROA	VITRO, S.A. DE C.V.
54	WALMEXV	WAL - MART DE MEXICO, S.A. DE C.V.

Tabla 5.14 Universo de activos riesgosos disponible.

Para seleccionar los diez activos con riesgo que debían ser incluidos en la cartera de inversión, se ejecutó en diez ocasiones el algoritmo genético 1; en cada corrida se utilizaron 250 cromosomas, 500 iteraciones, probabilidad de cruzamiento de 0.39 y

probabilidad de mutación de 0.05. La mejor aptitud alcanzada se logró en la corrida 10 donde el cromosoma $u = (11,15,18,22,29,39,40,42,49,52)$ obtuvo una aptitud de 46.77379578 en 441.4220 segundos. De esta forma, los activos seleccionados fueron:

Activo	Clave	Rendimiento	Riesgo
11	1. CMRB	0.00184302321867	0.03577622186192
15	2. EKCO*	0.00407312738435	0.03164272491188
18	3. FRAGUAB	0.00309278232601	0.01215787280806
22	4. GFINBURO	0.00146310508384	0.01807548933888
29	5. HILASALA	0.00058106077959	0.02783815737461
39	6. MOVILAB	0.00470407488310	0.02796231447699
40	7. NADROB	0.00028203817788	0.02176265530754
42	8. POSADASL	0.00153662854444	0.01690372557344
49	9. TMMA	0.00003145655838	0.02869482810572
52	10. VALLEB	0.00255727753865	0.02920232816145

Tabla 5.15 Activos seleccionados.

Una vez seleccionados los activos, se ejecutó el algoritmo genético 4 en 5 ocasiones, para encontrar los pesos que definen la cartera de mercado. Los parámetros utilizados fueron 250 cromosomas, 500 iteraciones, probabilidad de cruzamiento de 0.07, probabilidad de mutación de 0.02 y rendimiento del activo libre de riesgo (Cetes a 28 días) de 0.000183.

El mejor resultado se obtuvo en la corrida 3; en ella, el cromosoma $p = (0.0432, 0.0898, 0.3439, 0.0996, 0.0336, 0.1187, 0.0333, 0.1297, 0.0222, 0.0860)$ fue encontrado en 750.2030 segundos, mostrando una aptitud de 0.37819174. La cartera de inversión definida por p tiene rendimiento esperado de 0.00266193267870 y riesgo de 0.00655469796709.

Considerando el grado de aversión al riesgo del inversionista ($A = 100$), se ejecutó 5 veces el algoritmo genético 5. Los parámetros considerados fueron el rendimiento esperado y riesgo de la cartera definida por p , el rendimiento del activo libre de riesgo, $A=100$, 100 cromosomas, 150 iteraciones, probabilidad de cruzamiento de 0.01 y probabilidad de mutación de 0.005.

El mejor cromosoma encontrado, $y = 0.5770$, se encontró en las corridas 1 y 2 con tiempos de 18.0780 y 18.0150 segundos; la aptitud alcanzada por y fue de 0.0008981.

Considerando los resultados encontrados por los algoritmos genéticos 1, 4 y 5, la cartera óptima del inversionista está constituida aproximadamente de la siguiente forma: 2.49% de CMRB, 5.18% de ECO*, 19.84% de FRAGUAB, 5.75% de GFINBURO, 1.94% de HILASALA, 6.85% de MOVILAB, 1.92% de NADROB, 7.48% de POSADASL, 1.28% de TMMA, 4.96% de VALLEB y 42.3% Cetes a 28 días. Esta cartera, llamada CO1, tiene rendimiento diario esperado de 0.1613% y riesgo de 0.003782.

5.4.2 Administración de la cartera de inversión

Para la administración de la cartera de inversión, se considera que el inversionista tiene un capital inicial de \$1,000,000.00. La cartera de inversión CO1 debe ser comprada el 7 de junio de 2004, debido a que se le creó con la información registrada hasta el 4 de junio de 2004. En la tabla siguiente, se muestran los activos considerados en la cartera de inversión CO1 y la distribución que se realizó para su conformación basándose en los resultados de los algoritmos genéticos 1, 4 y 5.

07-Jun-04	Ponderación	Ponderación con ALR	Monto asignado	Precio del activo	Unidades por activo	Monto invertido
1. CMRB	0.0432	0.0249	\$ 24,926.40	\$ 1.05	23739	\$ 24,925.95
2. EKCO*	0.0898	0.0518	\$ 51,814.60	\$ 2.45	21148	\$ 51,812.60
3. FRAGUAB	0.3439	0.1984	\$ 198,430.30	\$ 31.00	6400	\$198,400.00
4. GFINBURO	0.0996	0.0575	\$ 57,469.20	\$ 15.20	3780	\$ 57,456.00
5. HILASALA	0.0336	0.0194	\$ 19,387.20	\$ 1.00	19387	\$ 19,387.00
6. MOVILAB	0.1187	0.0685	\$ 68,489.90	\$ 38.00	1802	\$ 68,476.00
7. NADROB	0.0333	0.0192	\$ 19,214.10	\$ 6.10	3149	\$ 19,208.90
8. POSADASL	0.1297	0.0748	\$ 74,836.90	\$ 6.31	11860	\$ 74,836.60
9. TMMA	0.0222	0.0128	\$ 12,809.40	\$ 29.00	441	\$ 12,789.00
10. VALLEB	0.0860	0.0496	\$ 49,622.00	\$ 18.00	2756	\$ 49,608.00
11. CETES 28		0.4230	\$ 423,000.00	\$ 10.00	42300	\$423,000.00
Subtotal (ACR)	1.0000	0.5770	\$ 577,000.00	\$ 148.11	94462	\$576,900.05
Subtotal (ALR)		0.4230	\$ 423,000.00	\$ 10.00	42300	\$423,000.00
Total	1.0000	1.0000	\$1,000,000.00	\$ 158.11	136762	\$999,900.05

Tabla 5.16 Cartera de inversión al 7 de junio de 2004.

En la primera columna de la tabla 5.16 se observan los activos considerados para conformar la cartera de inversión CO1; en la segunda columna se muestran las ponderaciones originalmente encontradas; en la tercera, se muestran las ponderaciones asignadas una vez que se considera la inclusión del activo libre de riesgo (ALR). En la cuarta columna se muestra el monto asignado a cada activo; en la quinta el precio por unidad de activo; en la sexta columna, el número de unidades que se pueden comprar por activo; y en la última columna, el monto realmente invertido en cada activo.

Así pues, el 7 de junio de 2004, el inversionista ocupa \$999,900.05 para comprar la cartera de inversión CO1, quedándose con un monto libre de \$99.95. La cartera de inversión CO1 es conservada por el inversionista hasta el día 18 de junio de 2004, día en que vende la cartera de inversión CO1.

18-Jun-04	Unidades por activo	Precio del activo	Monto
1. CMRB	23739	\$ 1.00	\$ 23,739.00
2. EKCO*	21148	\$ 3.01	\$ 63,655.48
3. FRAGUAB	6400	\$ 30.00	\$ 192,000.00
4. GFINBURO	3780	\$ 14.74	\$ 55,717.20
5. HILASALA	19387	\$ 1.00	\$ 19,387.00
6. MOVILAB	1802	\$ 35.00	\$ 63,070.00
7. NADROB	3149	\$ 6.10	\$ 19,208.90
8. POSADASL	11860	\$ 6.03	\$ 71,515.80
9. TMMA	441	\$ 29.00	\$ 12,789.00
10. VALLEB	2756	\$ 18.95	\$ 52,226.20
11. CETES 28	42300	\$ 10.03	\$ 424,080.77
Subtotal (ACR)	94462	\$ 144.83	\$ 573,308.58
Subtotal (ALR)	42300	\$ 10.03	\$ 424,080.77
Total	136762	\$ 154.86	\$ 997,389.35

Tabla 5.17 Cartera de inversión al 18 de junio de 2004.

Para el 18 de junio de 2004, el inversionista obtuvo \$997,389.35 por haber vendido la cartera CO1, que junto al monto libre de \$99.95 que poseía, suma \$997,489.30. Este nuevo monto lo utilizará el día 21 de junio de 2004 para comprar la nueva cartera de inversión CO2, que será conformada según los resultados que se obtengan a partir de las

ejecuciones de los algoritmos genéticos 4 y 5. En esta ocasión, el algoritmo genético 4 se ejecuta con una base de datos que ha sido aumentada en 10 rendimientos por cada acción considerada; y el algoritmo genético 5, se ejecuta considerando que para esa fecha el activo libre de riesgo otorga un rendimiento diario de 0.000182. Basándose en los resultados obtenidos y manteniendo su distribución estratégica, el inversionista realiza una nueva distribución táctica.

A partir de los resultados obtenidos en las corridas de los algoritmos genéticos 4 y 5, se elabora la cartera de inversión CO2 y se realizan los movimientos registrados en la tabla siguiente para lograr su adquisición el día 21 de junio de 2004.

21-Jun-04	Ponderación	Ponderación con ALR	Monto asignado	Precio del activo	Unidades por activo	Monto invertido
1. CMRB	0.0266	0.0127	\$ 12,674.92	\$ 1.05	12071	\$ 12,674.55
2. EKCO*	0.1088	0.0520	\$ 51,843.27	\$ 3.00	17281	\$ 51,843.00
3. FRAGUAB	0.3042	0.1453	\$ 144,951.49	\$ 30.00	4831	\$144,930.00
4. GFINBURO	0.1142	0.0546	\$ 54,416.37	\$ 14.90	3652	\$ 54,414.80
5. HILASALA	0.0392	0.0187	\$ 18,678.82	\$ 1.00	18678	\$ 18,678.00
6. MOVILAB	0.1523	0.0728	\$ 72,571.05	\$ 35.00	2073	\$ 72,555.00
7. NADROB	0.0283	0.0135	\$ 13,484.97	\$ 6.10	2210	\$ 13,481.00
8. POSADASL	0.1257	0.0600	\$ 59,896.13	\$ 6.03	9933	\$ 59,895.99
9. TMMA	0.0248	0.0118	\$ 11,817.22	\$ 29.01	407	\$ 11,807.07
10. VALLEB	0.0759	0.0363	\$ 36,166.40	\$ 18.95	1908	\$ 36,156.60
11. CETES 28		0.5223	\$ 520,988.66	\$ 10.00	52098	\$520,980.00
Subtotal (ACR)	1.0000	0.4777	\$ 476,500.64	\$ 145.04	73044	\$476,436.01
Subtotal (ALR)		0.5223	\$ 520,988.66	\$ 10.00	52098	\$520,980.00
TOTAL	1.0000	1.0000	\$ 997,489.30	\$ 155.04	125142	\$997,416.01

Tabla 5.18 Cartera de inversión al 21 de junio de 2004.

De acuerdo a los movimientos realizados para adquirir la cartera de inversión CO2, el inversionista cuenta con un monto libre de \$73.29. La cartera de inversión CO2, es conservada por el inversionista desde el 21 de junio de 2004 hasta el 2 de julio de 2004, día en que es vendida.

Los resultados obtenidos por la cartera de inversión CO2, se muestran en la tabla siguiente.

02-Jul-04	Unidades por activo	Precio del activo	Monto
1. CMRB	12071	\$ 1.10	\$ 13,278.10
2. EKCO*	17281	\$ 3.00	\$ 51,843.00
3. FRAGUAB	4831	\$ 29.00	\$ 140,099.00
4. GFINBURO	3652	\$ 16.50	\$ 60,258.00
5. HILASALA	18678	\$ 1.04	\$ 19,425.12
6. MOVILAB	2073	\$ 35.00	\$ 72,555.00
7. NADROB	2210	\$ 6.10	\$ 13,481.00
8. POSADASL	9933	\$ 6.03	\$ 59,895.99
9. TMMA	407	\$ 30.00	\$ 12,210.00
10. VALLEB	1908	\$ 17.00	\$ 32,436.00
11. CETES 28	52098	\$ 10.03	\$ 522,305.03
Subtotal (ACR)	73044	\$ 144.77	\$ 475,481.21
Subtotal (ALR)	52098	\$ 10.03	\$ 522,305.03
Total	125142	\$ 154.80	\$ 997,786.24

Tabla 5.19 Cartera de inversión al 2 de julio de 2004.

De esta forma, el día 2 de julio de 2004 el inversionista recibe \$997,786.24 que junto con el monto libre de \$73.29, suman \$997,859.53.

5.4.3 Resultados obtenidos

Se llevó a cabo la creación y administración de la cartera de inversión CO para un inversionista con grado de aversión al riesgo $A = 100$, basándose en los algoritmos genéticos 1, 4 y 5.

Por un lado, la creación consistió en la definición de la distribución estratégica y una primera distribución táctica. Para ello, primero se ejecutó el algoritmo genético 1, con una base de datos compuesta por los rendimientos diarios otorgados por 54 acciones cotizantes en la Bolsa Mexicana de Valores, durante el periodo comprendido entre el 4 de agosto de 2003 y el 4 de junio de 2004. A continuación se ejecutaron los algoritmos genéticos 4 y 5, que indicaron la distribución táctica que debía realizarse, considerando el rendimiento diario de un activo libre de riesgo (Cetes 28 días) y el grado de aversión al riesgo del inversionista.

Por otro lado, la administración de la cartera de inversión se realizó ejecutando nuevamente los algoritmos genéticos 4 y 5, pero considerando una base de datos actualizada en la que se cuenta con los rendimientos diarios generados por las acciones durante el primer lapso de la posesión de la cartera de inversión CO, así como el rendimiento diario que para entonces otorgó el activo libre de riesgo.

Tras un mes de administración de la cartera de inversión CO, se encontró que el proceso representó para el inversionista una pérdida de \$2140.47 ó 0.2140%. Obviamente, un resultado como estos jamás será bienvenido por el inversionista; sin embargo, antes de dar un dictamen final, es necesario considerar el comportamiento del mercado durante el periodo en que se creó y administró la cartera de inversión CO.

5.4.4 Comportamiento del mercado

A lo largo del mes de junio diversos factores de carácter político y económico tuvieron gran injerencia en el comportamiento de la Bolsa Mexicana de Valores; haciendo un análisis retrospectivo breve, se comentarán algunos de ellos.

El día lunes 7 de junio de 2004, día en que el inversionista compró los activos que conformarían la cartera de inversión CO1, se esperaban las mejores tendencias positivas para los mercados de Estados Unidos y en consecuencia para la Bolsa Mexicana de Valores. Las expectativas no defraudaron y la BMV registró su mejor jornada en lo que había transcurrido del año, alcanzando una ganancia de casi 18%. Este repunte fue atribuido principalmente a la baja de los precios del petróleo.

Para el resto de esa primera semana de la administración de la cartera de inversión CO1, se presentó una tendencia a la baja en el mercado. Los días martes y miércoles esta tendencia a la baja, fue consecuencia de la toma de utilidades después del repunte que ocurrió el día lunes 7 de junio; mientras que los días jueves y viernes, se

atribuyó a dos factores: el primero, los comentarios del presidente de la Reserva Federal de los EUA, Alan Greenspan, sobre una posible alza de las tasas de interés por motivo del informe del nivel de inflación registrado durante mayo en los EUA; y segundo, el cierre de los mercados el día viernes, por motivo de los funerales del ex presidente de los Estados Unidos, Ronald Reagan.

Para el día lunes 14 de junio, inicio de la segunda semana de la administración de la cartera de inversión CO1, las cosas no se modificaron mucho y la BMV volvió a caer. Fue en el día martes, después de darse a conocer la inflación registrada en mayo en los EUA, que se recobró el optimismo. Los comentarios de Alan Greenspan, quién declaró sobre un cambio paulatino en la política monetaria expansiva (es decir, un aumento gradual en las tasas de interés), fortalecieron el optimismo. No obstante, a partir del miércoles 16 de junio, la BMV nuevamente entró en un periodo de incertidumbre que continuó por el resto de la segunda semana de administración de la cartera de inversión CO1, argumentándose principalmente temores de que la FED tomara medidas más enérgicas sobre las tasas de interés.

En lo que respecta a la tercera semana de administración de la cartera de inversión CO, primera de CO2, se puede decir que la BMV se mantuvo a la expectativa de información relacionada con la inflación en México de la primera quincena de junio que se anunciaría el jueves 24 de junio. La incertidumbre asumida en el mercado, se reflejó en una semana que cerró en cuatro de cinco jornadas a la baja, siendo únicamente el día jueves cuando el Índice de Precios y Cotizaciones (IPC) registró una ganancia de 0.44%.

Para la cuarta semana de administración de la cartera de inversión CO, segunda semana de administración de CO2, hubo principalmente tres factores que provocaron poca operatividad en los mercados. El primero fue la espera hasta el día 30 de junio, del tan

anunciado aumento a las tasas de interés por parte de la Reserva Federal de los EUA, que por primera vez después de cuatro años, elevó la tasa referencial de fondos federales en un cuarto de punto. El segundo factor que influyó en los mercados en esa semana, fue el cambio de gobierno en Irak que podría generar nuevos conflictos. Por último, la segunda mitad de esta última semana de administración de la cartera de inversión CO, en los mercados se notó una postura tímida por parte de los inversionistas que prefirieron prepararse para un fin de semana largo en los mercados de EUA, los cuales permanecerían cerrados debido a las celebraciones del día de independencia de aquel país.

Este análisis breve que se ha realizado sobre el mercado en el periodo de administración de la cartera de inversión CO, no pretende justificar el resultado negativo obtenido; simplemente trata de mostrar que el marco contextual en el que se desarrolló el proceso de creación y administración de la cartera de inversión CO, no fue el más benévolo que se podía haber encontrado. Además, este análisis contribuye a formar un ejemplo más apegado a la realidad, que impide identificarlo como un ejemplo simplemente numérico.

6. CONCLUSIONES

En esta tesis se ha desarrollado un procedimiento para crear y administrar carteras de inversión, sin considerar la posibilidad de realizar ventas en corto, mediante algoritmos genéticos.

Una cartera de inversión es un conjunto de activos financieros que son combinados en iguales o distintas proporciones, con el objetivo de obtener lucro, mediante su posesión durante un determinado periodo de tiempo.

A cada activo financiero que existe en el mercado, se le puede asignar una medida de riesgo y una de rentabilidad esperada; ambas medidas se basan en el comportamiento histórico que el activo financiero ha descrito a lo largo del tiempo. Dentro de estos activos, existe uno con la particularidad de tener riesgo cero; es conocido como activo libre de riesgo y representa la obligación directa e incondicional del Gobierno de una nación que pagará una suma de dinero en una fecha específica.

A partir de esas dos características que posee cada activo, una cartera de inversión posee un rendimiento esperado y un nivel de riesgo. Basándose en esas dos medidas de una cartera de inversión, Harry Markowitz creó un modelo matemático que permite a un inversionista discernir entre buenas y malas carteras de inversión; a las buenas carteras de inversión, Markowitz las llamó carteras eficientes.

Una cartera de inversión eficiente es aquella que para un nivel de riesgo dado, maximiza el rendimiento esperado; ó desde otra perspectiva, para un rendimiento esperado dado, una cartera eficiente es aquella que minimiza el riesgo.

Con base en la cartera eficiente definida una vez que se han considerado activos con riesgo y el activo libre de riesgo, sólo hace falta considerar el grado de aversión al riesgo que posee un inversionista, para encontrar su cartera de inversión óptima.

El grado de aversión al riesgo de un inversionista está definido por su postura ante el riesgo. Un inversionista adverso al riesgo requiere de un incremento mayor en el rendimiento para aceptar un poco más de riesgo; un inversionista indiferente al riesgo no explica su comportamiento con base en los niveles de riesgo por asumir; y un inversionista propenso al riesgo está dispuesto a asumir una mayor cantidad de riesgo con tal de obtener un rendimiento adicional.

De acuerdo a los anteriores elementos, un inversionista crea su cartera de inversión óptima que administrará durante un periodo de tiempo con el incentivo de adquirir una ganancia monetaria al finalizar dicho periodo; esa ganancia estará dada por la diferencia entre los precios de los activos que componen su cartera, al inicio del periodo y al finalizar el periodo.

Los algoritmos genéticos son una técnica de optimización comprendida dentro de la computación evolutiva. Esta técnica de optimización, basándose en los principios de selección natural y de genética, plantean un mecanismo de búsqueda adaptativa de soluciones óptimas sobre el espacio de soluciones de un problema.

La estructura de los algoritmos genéticos es sumamente simple ya que está definida por cinco pasos: 1) se crea aleatoriamente una población de soluciones potenciales al problema por optimizar; 2) mediante una función de aptitud, que permita reflejar los criterios que distinguen a una solución buena del problema, se evalúa a cada miembro de la población para otorgarle un valor de aptitud; 3) basándose en los resultados obtenidos en el paso 2, se facilita la permanencia de las soluciones buenas y se dificulta la permanencia de las soluciones malas; 4) se permite el intercambio de información y se agrega variabilidad entre las soluciones que permanecen en la población para dar lugar a una nueva población,

todo ello se realiza mediante la aplicación de operadores genéticos; y 5) se regresa al paso dos para iniciar una nueva iteración.

El punto clave en la utilización de los algoritmos genéticos, es la forma en que se representa a un problema para que pueda ser resuelto por medio de esta técnica. En otras palabras, es necesario encontrar una codificación que permita representar a las soluciones del problema y que posibilite la manipulación de dichas soluciones en la forma más sencilla posible. La técnica de codificación más utilizada es la binaria.

Contando con una codificación de soluciones, los algoritmos genéticos indagan en el espacio de soluciones del problema para encontrar aquella que alcance una mayor calificación; esa calificación no es más que la cantidad numérica otorgada por la función de aptitud.

La técnica de los algoritmos genéticos está matemáticamente fundamentada en el Teorema del Esquema, desarrollado por John Holland quién es generalmente reconocido como el creador de los algoritmos genéticos. En dicho teorema se estipula que los algoritmos genéticos exploran el espacio de soluciones de un problema a partir de un tipo de estructuras peculiares llamadas esquemas. Los esquemas son representaciones de soluciones potenciales que facilitan el intercambio de información entre ellos mediante su utilización subsecuente y la aplicación de los operadores genéticos implementados.

A partir de esta técnica de optimización, se programaron 5 algoritmos genéticos para crear y administrar carteras de inversión. Cada uno de ellos tiene una función específica dentro del proceso por realizar, pero en algunos casos, los resultados de un algoritmo son el punto de partida de otro. Como base de datos solamente se requiere tener una matriz de rendimientos de los activos considerados en el universo de activos disponibles del inversionista; a pesar de que los algoritmos son capaces de operar sin

importar la cantidad de rendimientos considerados por cada activo, lo más recomendable es ejecutarlos con una base de datos que esté integrada por una cantidad considerable de rendimientos por cada activo, con el objetivo de eliminar la posibilidad de obtener resultados insatisfactorios debido a la escasez de información. En lo que se refiere a los operadores genéticos utilizados, cada algoritmo aplica los operadores de cruzamiento y mutación; a estos operadores se suma el criterio elitista incorporado en cada algoritmo para alcanzar las mejores soluciones posibles durante cada corrida ejecutada. Además, en cada algoritmo genético se prefirió programar como condición de terminación el número de iteraciones por realizar; de esta forma, los algoritmos no se ejecutan hasta lograr la convergencia de la población porque esta no muy frecuentemente es alcanzada.

El primero de ellos, único programa que opera con base en una codificación de números enteros, es el encargado de seleccionar los activos con riesgo que deberán integrar la cartera de inversión. En otras palabras, la distribución estratégica de las carteras de inversión creadas a partir de estos algoritmos genéticos, queda definida únicamente a partir del algoritmo genético 1.

La función de aptitud utilizada en este primer algoritmo, en un intento muy ambicioso por considerar los rendimientos esperados, los riesgos y los coeficientes de correlación involucrados, favorece la selección de un conjunto de activos con buena diversificación, pero puede dejar poco satisfecho al inversionista por los rendimientos correspondientes.

Es precisamente la forma de la función de aptitud del primer algoritmo genético, el elemento que debe ser ajustado para representar los objetivos que se pretenden alcanzar con la cartera de inversión que se está creando. Por ejemplo, si se quiere formar una cartera de inversión que busque ser atractiva en rendimiento, la función de aptitud deberá depender

en mucho mayor medida de los rendimientos considerados y en menor medida de las desviaciones estándar y los coeficientes de correlación.

El algoritmo genético 2, programado en codificación binaria al igual que el resto de los algoritmos genéticos, es el encargado de encontrar la cartera eficiente que presente el mayor cociente rendimiento esperado – varianza; fundamentado en el objetivo antes mencionado, su función de aptitud esta dada como al cociente entre el rendimiento esperado de la cartera de inversión evaluada y la varianza de la misma. En este caso los activos considerados son únicamente aquellos que fueron seleccionados mediante el algoritmo genético 1.

El algoritmo genético 3 es el encargado de encontrar el punto de varianza mínima, considerando los activos con riesgo previamente seleccionados por el primero de los algoritmos genéticos. Para ello, la función de aptitud depende sólo de la desviación estándar o riesgo que posee la cartera de inversión evaluada.

Los algoritmos genéticos 2 y 3, operativamente, no tienen tanta importancia para el inversionista. Sin embargo, los resultados obtenidos a partir de ellos, pueden significar carteras de referencia o simplemente datos que permitan obtener un panorama más claro del comportamiento de las carteras de inversión que se pueden crear a partir de los activos previamente seleccionados.

El algoritmo genético 4 se encarga de identificar los pesos por asignar para crear la cartera de mercado que define, junto con el activo libre de riesgos, la recta tangente a la región originalmente factible, también llamada línea de asignación de capitales. La forma adoptada por su función de aptitud, se desprende de un sencillo análisis geométrico en el que se desea maximizar la pendiente de la recta tangente definida por el activo libre de riesgo y la región factible generada a partir de los activos con riesgo considerados.

El último de los algoritmos genéticos elaborado, considerablemente más sencillo para programar que los anteriores, se encarga de encontrar la cartera de inversión que se ubica sobre la recta tangente definida por la cartera de inversión encontrada en el algoritmo genético 4 y el activo libre de riesgo, tomando en cuenta el grado de aversión al riesgo que presenta el inversionista.

Como se puede observar, los algoritmos genéticos 2, 3, 4 y 5 realizan una distribución táctica pero considerando diferentes criterios; es decir, asignan ponderaciones a cada uno de los activos considerados para conformar una cartera de inversión con alguna característica en particular.

Así pues, el proceso de creación de una cartera de inversión se realiza con base en los resultados obtenidos a partir de las ejecuciones de los algoritmos genéticos 1, 4 y 5, principalmente. Para llevar a cabo la administración de la misma, es necesario ejecutar los algoritmos genéticos 4 y 5, tan frecuentemente como se quiera definir una nueva distribución táctica.

Cada que se desee crear una nueva cartera de inversión, deberá ser ejecutado el algoritmo genético 1 para definir la distribución estratégica y los algoritmos genéticos 2, 3, 4 ó 5, dependiendo del tipo de distribución táctica que se requiera en la correspondiente cartera de inversión.

Para hablar sobre la injerencia de los parámetros de entrada en los algoritmos genéticos programados, se les dividió en parámetros de carácter financiero y parámetros de carácter genético. Sobre los parámetros de carácter financiero, se prefirió no realizar un análisis debido a que éstos dependen principalmente de la información a la que tenga acceso el inversionista. Sobre los parámetros de carácter genético (número de cromosomas por considerar en cada generación, número de iteraciones por realizar en cada corrida,

probabilidad de aplicación del operador cruzamiento y probabilidad de aplicación del operador mutación), los resultados otorgados por los algoritmos genéticos mostraron mayor dependencia con el parámetro de número de iteraciones.

Un punto importante por resaltar relacionado con lo expuesto en el párrafo anterior, es el comportamiento mostrado por los algoritmos genéticos que permite señalar que la diversidad de población y precisión selectiva están fuertemente ligados. Como muestra de ello, se observó que un incremento en la precisión selectiva disminuye la diversidad de la población y viceversa; así, en algunas ocasiones una fuerte precisión selectiva hacía inefectiva la búsqueda.

En lo que respecta a los resultados obtenidos en la aplicación de los algoritmos genéticos programados, se puede afirmar que, en conjunto, resultaron ser una excelente herramienta para crear carteras de inversión; el ejemplo en el que se considera un universo de activos disponibles reducido, avala en parte esta aseveración. Sin embargo, en lo que se refiere a la administración de la cartera de inversión, los resultados no fueron los más alentadores.

También se puede señalar que los algoritmos genéticos son una técnica de optimización que representan una buena alternativa para ser aplicada en el ámbito financiero. Representan una herramienta potente y sumamente moldeable para cumplir con los requerimientos de un problema financiero. Como muestra de esta sorprendente capacidad de adaptación, se tienen los algoritmos genéticos 2, 3 y 4, los cuales son sumamente parecidos pero cada uno otorga la solución de una variante del problema financiero considerado.

Otra ventaja que se pudo comprobar con los algoritmos genéticos elaborados, es la facilidad con que se les puede programar. Observando los códigos de cada uno de ellos,

salta a la vista la enorme simplicidad con que fueron contruidos; en ninguno de ellos se utilizaron conocimientos avanzados de programación y, a pesar de ello, los resultados pueden considerarse satisfactorios.

Por último, vale la pena mencionar que los algoritmos programados en este trabajo, podría ser vistos como una versión preliminar de una versión más potente. Esto se debe a que los códigos programados pueden ser mejorados en aras de alcanzar resultados más precisos y en un menor tiempo. La razón por la que en la elaboración de éstos códigos no se tuvieron esos objetivos como principales, fue debido a que se deseaba en primera instancia, comprobar la facilidad de programación y corroborar que, efectivamente sin un conocimiento tan profundo de programación en un lenguaje en particular, se podían alcanzar resultados satisfactorios.

NOTAS

1. Este texto es comprensible para toda aquella persona interesada en el tema de la *Teoría de Carteras de Inversión* y/o de los *Algoritmos Genéticos* que posea, al menos, conocimientos básicos en las áreas de finanzas, programación y probabilidad. Si el lector, además, posee conocimientos básicos de las áreas investigación de operaciones y economía, el texto le resultará completamente comprensible.
2. La decisión de administrar la cartera de inversión durante cuatro semanas, se debió a que la información disponible de los precios registrados por los activos financieros sobre los que se calcularon los correspondientes rendimientos, abarcan el periodo comprendido entre el 1° de agosto de 2003 y el 2 de julio de 2004 (235 rendimientos diarios por activo). A partir de esa información, se planeó la conformación de una base de datos inicial de 220 rendimientos diarios por activo; de esta forma quedó definido el periodo de cuatro semanas para realizar la administración de la cartera de inversión.
Algunas personas podrían considerar que un periodo de administración de cuatro semanas no es suficiente para evaluar los resultados; sin embargo, es importante considerar que la inversión en bolsa no está restringida al largo plazo y que un buen largo plazo está integrado por buenos cortos plazos.
3. MATLAB es el nombre abreviado de "MATrix LABoratory". MATLAB es un programa para realizar cálculos numéricos con vectores y matrices. Como caso particular puede también trabajar con números escalares, tanto reales como complejos. Tiene la capacidad de realizar una amplia variedad de gráficos en dos y tres dimensiones. MATLAB tiene también un lenguaje de programación propio. La versión 6.0 es la utilizada en este trabajo.
MATLAB trabaja siempre en doble precisión, es decir guardando cada dato en 8 bytes, con unas 15 cifras decimales exactas. También puede trabajar con cadenas de caracteres (strings) y también con otros tipos de datos: matrices de más dos dimensiones, vectores y matrices de celdas, estructuras y clases y objetos.
MATLAB tiene un gran número de funciones incorporadas. Algunas son funciones intrínsecas, esto es, funciones incorporadas en el propio código ejecutable del programa. Estas funciones son particularmente rápidas y eficientes. Existen además funciones definidas en ficheros *.m que vienen con el propio programa o que han sido aportadas por usuarios del mismo. Estas funciones extienden las posibilidades del programa.
4. En los mercados bursátiles, como en la vida, es imposible eliminar la incertidumbre sobre lo que sucederá en el futuro. Así, a pesar de que existen métodos que proporcionan una idea confiable sobre el valor que asumirá el rendimiento de un activo financiero en una fecha futura, no se puede solicitar sistemáticamente un rendimiento determinado porque esos métodos son sólo estimaciones calculadas a priori con un cierto margen de error.
5. Supóngase que se desea calcular el rendimiento anual de un activo financiero A con base en los rendimientos obtenidos cada mes. El rendimiento anual es el producto de los rendimientos mensuales; esto se debe a que se trata de una situación similar a la del interés compuesto que se devenga cada mes. Así, el rendimiento del segundo

mes considerado, se aplica sobre el principal de la inversión y sobre el interés del primer mes.

Entonces el rendimiento del activo A es:

$$r_A = \left(\frac{P_1}{P_0} \times \frac{P_2}{P_1} \times \dots \times \frac{P_{12}}{P_{11}} \right) - 1 = (1+r_1) \times (1+r_2) \times \dots \times (1+r_{12}) - 1 \quad (1)$$

Aplicando logaritmo natural a (1) se tiene:

$$\ln\left(\frac{P_1}{P_0}\right) + \ln\left(\frac{P_2}{P_1}\right) + \dots + \ln\left(\frac{P_{12}}{P_{11}}\right) = \ln(1+r_1) + \ln(1+r_2) + \dots + \ln(1+r_{12}) = r_A^c \quad (2)$$

En (2), r_A^c denota el rendimiento continuo del activo A ; es decir, aquella tasa que compuesta continuamente es la tasa simple anual. Por lo tanto:

$$r_A^c = \ln\left(\frac{P_{12}}{P_0}\right) = \ln(1+r_A) \quad (3)$$

La primera igualdad de (3) en su forma general se escribe del modo siguiente:

$$r_A^c = \ln\left(\frac{P_{t+n}}{P_t}\right)$$

6. Cuando se considera que la mejor estimación sobre el rendimiento futuro es el rendimiento histórico se asume que el mercado bursátil es un mercado eficiente. Dos de las definiciones más aceptadas de mercado eficiente son las siguientes:
 - a) Un mercado bursátil es eficiente si cuenta con eficiencia interna y eficiencia externa. La eficiencia interna se entiende como la capacidad que tienen los precios de los activos financieros para ajustarse rápidamente a las señales del mercado. La eficiencia externa se entiende como la capacidad que tienen los precios de los activos financieros para ajustarse rápidamente a las señales de la economía.
 - b) Un mercado bursátil es eficiente si los precios de los activos financieros se ajustan a la información del pasado, del presente y a las expectativas del futuro.
7. *Broker* es el término anglosajón que se utiliza para referirse a la persona física o jurídica que actúa como intermediario financiero en los mercados financieros; en otras palabras, es el ente que compra y vende valores por cuenta de sus clientes.
8. Para ello se asume que los rendimientos de los activos financieros son variables aleatorias mutuamente independientes y tienen varianzas constantes (homocedasticidad); entonces, según el Teorema de Límite Central, la distribución de probabilidades de la variable aleatoria r_t sigue la distribución Normal. Así, el valor esperado $E(r_t)$, como el rendimiento del activo, y su varianza $\sigma^2(r_t)$, como medida de riesgo asociada al activo, son suficientes para describir la distribución del rendimiento futuro de la cartera.
9. El valor esperado es la primera característica numérica de una variable aleatoria (v.a.). También conocido como valor promedio, media o primer momento, se define como sigue:

- a) Si X es una v.a. discreta con valores x_0, x_1, x_2, \dots ; $E(X) = \sum_{i=0}^{\infty} x_i P(X = x_i)$.

b). Si X es una v.a. con una función de densidad f_x ; $E(X) = \int_{-\infty}^{\infty} xf_x(x) dx$.

El valor esperado, que es un promedio ponderado de los valores que puede asumir la variable aleatoria, tiene propiedades importantes. Siendo X y Y variables aleatorias y c constante, dos de las más comúnmente utilizadas se expresan de la siguiente forma: $E(cX) = cE(X)$ y $E(X + Y) = E(X) + E(Y)$. A partir de ellas, se afirma que el valor esperado es una función lineal.

10. El comportamiento de una acción depende directamente de la situación en que se encuentra inmersa la empresa emisora; esa situación depende de diversos factores como son: los estados financieros, los balances, los resultados de ventas, el flujo de efectivo, las políticas de dividendo e inversión, la capacidad gerencial y productiva, la competencia que existe en el sector, los proyectos de inversión, las perspectivas de crecimiento, el entorno político, el tipo de cambio, los tratados internacionales de comercio y la legislación, entre otros. Así, dado que cada empresa enfrenta una situación particular, cada activo se mueve de forma distinta en un mismo momento dentro del mercado.
11. El riesgo no sistemático se atribuye a los factores específicos de la empresa (resultados de ventas, capacidad gerencial y productiva, proyectos de inversión) y a los factores específicos del sector (competencia, demanda de materias primas, tecnología de producción necesaria en el sector). Así, el riesgo no sistemático en una cartera de inversión asumido como resultado de la inclusión de una acción cuya empresa emisora se está viendo afectada por factores negativos, puede ser compensado con la inclusión de una acción cuya empresa emisora se está viendo afectada por factores positivos.
12. Para el ejemplo se supuso $\sigma_{ij} = .22\sigma^2 \quad i \neq j$. El factor .22 fue elegido arbitrariamente; sin embargo, se puede considerar el mismo desarrollo matemático utilizando un factor $0 < \alpha \leq 1$ de tal forma que se concluiría con la expresión siguiente:

$$\text{var}(r) = \frac{(1-\alpha)\sigma^2}{n} + \alpha\sigma^2.$$

13. Se dice que dos activos están perfectamente correlacionados si su coeficiente de correlación (también conocido como coeficiente de correlación lineal) es idéntico a 1 ó -1; es decir, los activos a y b están perfectamente correlacionados si:

$$\rho = \frac{\text{cov}(a,b)}{\sigma_a\sigma_b} = 1 \quad \text{ó} \quad \rho = \frac{\text{cov}(a,b)}{\sigma_a\sigma_b} = -1$$

14. El generador de números aleatorios utilizado, determina los valores con base en una distribución uniforme sobre el intervalo (0,1).

APÉNDICE 1: CÓDIGOS DE LOS ALGORITMOS GENÉTICOS

Algoritmo genético 1

```
function [D1,e2,u3] = ag1_ele(M,na,nc,ni,pc,pm)
% Función:
% - ag1_ele es el nombre de la función que corre el algoritmo genético 1, elección de los activos
%   financieros, utilizando como operadores genéticos el cruzamiento y la mutación.
% Datos:
% - M es la matriz que contiene las series históricas de los rendimientos de los activos por considerar,
%   cada serie histórica ocupa una columna de la matriz.
% - na es el número de activos que conformarán la cartera de inversión.
% - nc es el número de cromosomas o individuos de la población.
% - ni es el número máximo de iteraciones.
% - pc es la probabilidad de cruzamiento.
% - pm es la probabilidad de mutación.
% Resultados:
% - D1 es la matriz de la última generación.
% - e2 es el cromosoma con mayor aptitud de la última generación.
% - u3 es el valor de aptitud del cromosoma e2.
c = size(M);           % Se calculan las dimensiones de M.
r = vec_ren(M);       % Se calculan el vector de rendimientos, el vector de desviaciones estándar y
d = vec_des(M);       % la matriz de coeficientes de correlación.
C = mat_cor(M);
t = 0;                % Se inicializa contador de iteraciones.
P = pob_ini(nc,c(1,2)); % Se genera población inicial.
while t < ni          % Se inician las iteraciones del algoritmo genético.
    t = t + 1;        % Se actualiza el contador de iteraciones.
    e = zeros(1,c(1,2)); % Se crea un vector para ubicar al mejor cromosoma.
    e1 = 0;           % Se crea una variable falso / verdadero.
    v = eva_pob(P,r,d,C,na); % Se evalúa la población actual.
    u1 = max(v);      % Se identifica la mejor aptitud en la población actual.
    for i = 1:nc      % Se ubica la posición de la mejor aptitud en la población actual de acuerdo
        if v(1,i) == u1 % al vector de evaluaciones y se copia el cromosoma correspondiente de P
            if e(1,1) == 0 % en el vector e. En caso de estar repetido el mejor cromosoma, sólo se copia
                e(1,:) = P(i,:); % una vez; para la segunda vez que se quiera copiar, e(1,1) no será igual a cero.
            else
                e = e;
            end
        else
            e = e;
        end
    end
    Q = pob_sel(P,v); % Se seleccionan los individuos de P que podrán seguir adelante.
    R = ope_cru(Q,pc); % Se aplica el operador cruzamiento.
    P = ope_mut(R,pm); % Se aplica el operador mutación.
    v = eva_pob(P,r,d,C,na); % Se evalúa la población que se generó después de los operadores.
    u2 = min(v);        % Se identifica la peor aptitud de la población modificada.
    if u1 > u2          % Se aplica elitismo, se compara la aptitud del peor cromosoma
        for i = 1:nc % de la población modificada con la mejor aptitud de la
            if v(1,i) == u2 % población que inicio la iteración. En caso de que la aptitud
                if e1 == 0 % del mejor cromosoma de la primera población sea mejor que
                    P(i,:) = e(1,:); % la aptitud del peor cromosoma de la población modificada,
                    e1 = 1; % el primero sustituye al segundo en la población modificada;
                else % en caso contrario, la población modificada queda intacta.
                end
            end
        end
    end
end
```

```

        P = P;
    end
else
    P = P;
end
end
else
    P = P;
end
end
% Termina procedimiento general del algoritmo genético.
D = zeros(nc,na); % Se crea una matriz que contiene sólo los primeros na genes
for i=1:nc*na % de cada cromosoma de la última población de la corrida.
    D(i) = P(i);
end
D1 = zeros(nc,na); % Se crea la matriz que contiene sólo los primeros na genes
for i=1:nc % de cada cromosoma de la última población de la corrida,
    D1(i,:) = sort(D(i,:)); % pero con los genes ordenados.
end
e2 = zeros(1,na); % Se crea un vector para ubicar al mejor cromosoma en la corrida.
u3 = max(v); % Se identifica el máximo valor de aptitud obtenido en la corrida.
for i=1:nc % Se ubica al cromosoma que mostró el mayor valor de aptitud y
    if v(1,i)==u3 % se copia.
        e2 = D1(i,:);
    else
        e2 = e2;
    end
end
e2 % Se muestra el mejor cromosoma encontrado en la corrida.
u3 % Se muestra la aptitud del mejor cromosoma encontrado.

```

```

function [c] = cot_des(d,na)
% Función:
% - cot_des es el nombre de la función que calcula la suma de las na mayores
% desviaciones estándar disponibles a partir del conjunto de activos considerados.
% Datos:
% - d es el vector de desviaciones estándar de cada activo.
% - na es el número de activos que conformarán la cartera de inversión.
% Resultado:
% - c es la suma de las na desviaciones estándar mayores disponibles.
m = size(d); % Se calculan las dimensiones de d.
s = sort(d); % Se crea un nuevo vector ordenado a partir de d.
c = 0; % Se define variable para la cota.
for i=m(1,2):-1:m(1,2)-na+1 % Se calcula la cota.
    c = c+s(1,i);
end

```

```

function [c] = cot_ren(r,na)
% Función:
% - cot_ren es el nombre de la función que calcula la cota inferior del rendimiento para
% la posterior creación de la función de aptitud del algoritmo genético 1.
% Datos:
% - r es el vector de rendimientos promedio de cada activo.
% - na es el número de activos que conformarán la cartera de inversión.
% Resultado:
% - c es la suma de los peores na rendimientos disponibles.
s = sort(r); % Se crea un vector ordenado a partir de r.

```

```

c = 0;           % Se define variable para la cota.
for i=1:na      % Se calcula la cota.
    c = c+s(1,i);
end

function [v] = eva_pob(P,r,d,C,na)
% Función:
% - eva_pob es el nombre de la función que genera el vector que contiene la
%   evaluación de cada cromosoma de la población.
% Datos:
% - P es la matriz que contiene en cada renglón a un individuo de la población.
% - r es el vector de rendimientos promedio de cada activo.
% - d es el vector de desviaciones estándar de cada serie de rendimientos.
% - C es la matriz de correlaciones.
% - na es el número de activos que conformarán la cartera de inversión.
% Resultado:
% - v es el vector de evaluaciones de los cromosomas.
a = size(P);           % Se calculan las dimensiones de P.
v = zeros(1,a(1,1));  % Se crea el vector de evaluaciones.
b = cot_ren(r,na);    % Se calcula la cota de los rendimientos.
if b<0                % Se calcula valor absoluto de la cota de
    b = -b;           % los rendimientos.
else
    b = b;
end
for i=1:a(1,1)        % Se suman las constantes de la función de
    v(1,i) = b+cot_des(d,na)+((na-1)*na/2)+0.0001; % evaluación para cada cromosoma.
end
for i=1:a(1,1)        % Se agregan los rendimientos, las desviaciones
    for j=1:na        % estándar y los coeficientes de correlación
        v(1,i) = v(1,i)+r(1,P(i,j))-d(1,P(i,j)); % que completan la aptitud de cada cromosoma.
        for k=j+1:na
            v(1,i) = v(1,i)-C(P(i,j),P(i,k));
        end
    end
end
end

function [C] = mat_cor(M)
% Función:
% - mat_cor es el nombre de la función que genera la matriz de correlaciones.
% Dato:
% - M es la matriz que en cada columna contiene la serie de rendimientos de un activo.
% Resultado:
% - C es la matriz de correlaciones.
d = size(M);           % Se calculan las dimensiones de M.
C = zeros(d(1,2));    % Se crea la matriz cuadrada C.
N = zeros(2);         % Se crea una matriz de apoyo.
for i=1:d(1,2)        % Para cada par de activos considerados se calcula su matriz
    for j=1:d(1,2)    % de coeficientes de correlación N y se rescata el elemento
        N = corrcoef(M(:,i),M(:,j)); % correspondiente para ubicarlo en la matriz de
        C(i,j) = N(1,2); % coeficientes de correlación C que considera al total de
    end               % activos.
end
end

function [R] = ope_cru(Q,pc)
% Función:

```

```

% - ope_cru es el nombre de la función que aplica el operador cruzamiento a la
% población generada para cruzamiento.
% Datos:
% - Q es la matriz que contiene en cada renglón a un individuo de la población.
% - pc es la probabilidad de cruzamiento.
% Resultado
% - R es el matriz con la población creada después de la aplicación del operador cruzamiento.
c = size(Q); % Se calculan las dimensiones de Q.
R = zeros(c); % Se crea la matriz que contendrá la población cruzada.
r = rand(1,c(1,1)); % Se crea un vector de números aleatorios entre 0 y 1.
s = zeros(1,c(1,1)); % Se crea un vector indicador de cromosomas por cruzar.
for i=1:c(1,1) % Se realizan las comparaciones necesarias para
    if r(1,i)<pc % identificar a los cromosomas elegidos para cruzar;
        s(1,i) = 1; % un cromosoma por cruzar se identifica con un 1 en
    else % su correspondiente posición en el vector indicador s.
        s(1,i) = s(1,i);
    end
end
m = sum(s)/2; % Se calcula el número de cromosomas por cruzar por .5.
[m1,m2] = rat(m); % Se utiliza la función rat para conocer dos números a
% y b tales que (a/b)=m. Si b es igual a 1, el número de
% cromosomas por cruzar es par y queda intacto. En caso
% de que el número de cromosomas por cruzar sea impar,
% se genera un número aleatoriamente que indique el
% cromosoma que será agregado (o eliminado) para
% aplicar el operador cruzamiento.
if m2==1
    s = s;
else
    r1 = randperm(c(1,1));
    if s(1,r1(1,1))==0 % cromosoma que será agregado (o eliminado) para
        s(1,r1(1,1)) = 1; % aplicar el operador cruzamiento.
    else
        s(1,r1(1,1)) = 0;
    end
end
m3 = sum(s); % Se calcula el número par de cromosomas por cruzar.
if m3==0 % Si el numero de cromosomas por cruzar es 0, la
    R=Q; % población queda intacta.
else % Si el número de cromosomas por cruzar es mayor a 0,
    U = zeros(c(1,1),c(1,2)+1); % se crea una matriz de referencia U.
    for i=1:c(1,1) % Se numeran los renglones de U sobre la primera
        U(i,1) = i; % columna
    end
    for i=1:c(1,1) % Se copian la cromosomas por cruzar en su
        U(i,2:c(1,2)+1) = s(1,i)*Q(i,:); % correspondiente renglón y los que no se cruzarán
    end % se les hace renglones de ceros
    U1 = zeros(m3,c(1,2)+1); % Se crea una matriz U1 de cromosomas por cruzar.
    k = 1; % Se colocan en U1 solamente los cromosomas por cruzar,
    for i=1:c(1,1) % la primera columna de U1 también indica el renglón
        if U(i,2)==0 % que ocupaba el correspondiente cromosoma en la
            k = k; % matriz de la población original.
        else
            U1(k,:) = U(i,:);
            k = k+1;
        end
    end
    U2 = zeros(m3,c(1,2)+1); % Se crea una nueva matriz de apoyo para cruzar.
    for i=1:m3 % Se copia en U2 la primera columna de U1.
        U2(i,1) = U1(i,1);
    end
    for i=1:2:m3-1 % Se comienza a aplicar el cruzamiento por parejas.

```

```

p = randperm(c(1,2)-1); % La primera entrada del vector aleatorio indica el punto de cruzamiento.
U2(i,2:p(1,1)+1) = U1(i,2:p(1,1)+1); % Se copian los primeros p(1,1) genes del padre.
k1 = p(1,1)+1; % Se indica el punto donde debe continuar el operador.
for j=2:c(1,2)+1 % Se termina la aplicación del operador cruzamiento en
    if U1(i+1,j)~=U2(i,2:c(1,2)+1) % el primer cromosoma hijo, completándolo con
        U2(i,k1+1) = U1(i+1,j); % los que le hacen faltan, pero siguiendo el orden
        k1 = k1+1; % que estos guardan en el cromosoma madre.
    else
        U2 = U2;
    end
end
U2(i+1,2:p(1,1)+1) = U1(i+1,2:p(1,1)+1); % Se forma al segundo hijo de la misma
k2 = p(1,1)+1; % forma que se hizo para el primer hijo.
for j=2:c(1,2)+1
    if U1(i,j)~=U2(i+1,2:c(1,2)+1)
        U2(i+1,k2+1) = U1(i,j);
        k2 = k2+1;
    else
        U2 = U2;
    end
end
end
for i=1:m3 % Se colocan a los hijos formados por cruzamiento, en
    R(U2(i,1,:))=U2(i,2:c(1,2)+1); % la matriz R, ocupando la posición de sus padres
end
for i=1:c(1,1) % Se completa la matriz R con los cromosomas que no se
    if R(i,1)==0 % vieron afectados por la aplicación del operador
        R(i,:) = Q(i,:); % cruzamiento.
    else
        R = R;
    end
end
end
function [U] = ope_mut(R,pm)
% Función:
% - ope_mut es el nombre de la función que aplica el operador mutación a la población.
% Datos:
% - R es la matriz que contiene en cada renglón a un individuo de la población.
% - pm es la probabilidad de mutación.
% Resultado:
% - U es el matriz con la población después de la aplicación del operador mutación.
c = size(R); % Se calculan las dimensiones de R.
U = zeros(c); % Se crean la matriz U que será la población mutada.
S = zeros(c); % Se crea una matriz auxiliar.
T = rand(c); % Se crea una matriz de números aleatorios entre 0 y 1.
for i=1:c(1,1) % Se realizan las comparaciones correspondientes para
    for j=1:c(1,2) % identificar los genes que deben mutar, un gen que mutará
        if T(i,j)<pm % es marcado con 1 en su correspondiente posición en la
            S(i,j) = 1; % matriz S.
        else
            S = S;
        end
    end
end
end
m = zeros(1,c(1,1)); % Se crea un vector, cada entrada será asignada a un cromosoma.

```

```

for i=1:c(1,1) % Para cada cromosoma, se verifica que el número de genes por
    m(1,i) = sum(S(i,:))/2; % mutar en él, sea un número par. Si el número de genes por
    [m1,m2] = rat(m(1,i)); % mutar es par, el cromosoma queda intacto; si el número de
    if m2==1 % de genes por mutar es impar, se selecciona aleatoriamente
        S = S; % uno de sus genes y se le selecciona (o elimina) para
    else % aplicar el operador mutación.
        r = randperm(c(1,2));
        if S(i,r(1,1))==0
            S(i,r(1,1)) = 1;
        else
            S(i,r(1,1)) = 0;
        end
    end
end
S1 = zeros(c); % Se crea una segunda matriz auxiliar.
S1 = R * S; % Se colocan los genes por mutar en la matriz.
for i=1:c(1,1) % Se recorre la matriz S1 para ubicar los genes que mutarán,
    if S1(i,:)==0 % es decir las entradas cuyo valor sea distinto de cero. Una
        U = U; % vez que se ubica a un primer gen que mutará, se guarda su
    else % valor en una variable; se inicia la búsqueda del otro gen
        j = 1; % que será su pareja para aplicar el operador mutación.
        while j<c(1,2)+1 % Cuando ambos han sido ubicados, se les coloca en la matriz
            if S1(i,j)==0 % resultado U, pero intercambiándolos en posición. Se
                j = j+1; % prosigue de la misma forma hasta recorrer la matriz S1
            else % completamente.
                c1 = S1(i,j);
                k = j+1;
                while k<c(1,2)+1
                    if S1(i,k)==0
                        k = k+1;
                    else
                        c2 = S1(i,k);
                        U(i,j) = c2;
                        U(i,k) = c1;
                        j = k+1;
                        k = c(1,2)+1;
                    end
                end
            end
        end
    end
end
end
for i=1:c(1,1)*c(1,2) % Se completa la matriz U, tomando los elementos que no fueron
    if U(i)==0 % afectados por la aplicación del operador mutación de la
        U(i) = R(i); % matriz de la población original R.
    else
        U(i) = U(i);
    end
end
end

```

```
function [M]=pob_ini(tam_pob,num_gen)
```

```
% Función:
```

```
% - pob_ini es la función que genera la población inicial del algoritmo genético 1.
```

```
% Datos:
```

```
% - tam_pob es el número de individuos en la población.
```

```
% - num_gen es el número de genes de cada individuo en la población.
```

```

% Resultado:
% - M es una matriz en la que cada renglón representa a un individuo.
M = zeros(tam_pob,num_gen); % Se crea la matriz que contendrá a la población inicial.
for i = 1:tam_pob           % Para cada renglón se crea una permutación que
c = randperm(num_gen);     % representará a un cromosoma y se coloca en el
    for j = 1:num_gen      % respectivo renglón de la matriz M.
        M(i,j) = c(j);
    end
end

function [Q] = pob_sel(P,v)
% Función:
% - pob_sel es el nombre de la función que genera una nueva población a partir de la
% población actual para que posteriormente a esta nueva generación le sean
% aplicados los operadores considerados por el algoritmo genético.
% Datos:
% - P es la matriz que contiene en cada renglón a un individuo de la población.
% - v es el vector de evaluaciones.
% Resultado:
% - Q es el matriz formada por los individuos seleccionados de la generación actual para que les sean
% aplicados los operadores genéticos que darán como resultado a una nueva generación.
c = size(P); % Se calculan las dimensiones de P.
Q = zeros(c); % Se crea la matriz resultado Q.
f = sum(v); % Se calcula la suma de aptitudes.
p = zeros(1,c(1,1)); % Se crea un vector auxiliar.
for i=1:c(1,1) % Se calcula el vector que mostrara la función de
    p(1,i) = v(1,i)/f; % densidad de la población.
end
q = zeros(1,c(1,1)); % Se crea un vector auxiliar.
q(1,1) = p(1,1); % Se calcula el vector que mostrara la función de
for i=2:c(1,1) % distribución de la población.
    q(1,i) = q(1,i-1)+p(1,i);
end
r = rand(1,c(1,1)); % Se genera un vector de números aleatorios entre 0 y 1.
s = zeros(1,c(1,1)); % Se crea un vector de referencia.
for i=1:c(1,1) % Se realizan las comparaciones correspondientes para
    if r(1,i)<q(1,1) % ubicar a los cromosomas seleccionados. En cada
        s(1,i) = 1; % entrada del vector s, se colocará el número de
    else % renglón en el que se encuentra ubicado el
        for j=1:c(1,1)-1 % cromosoma elegido para la nueva población de
            if r(1,i)>q(1,j) & r(1,i)<q(1,j+1) % acuerdo a las comparaciones realizadas.
                s(1,i) = j+1;
            else
                s(1,i) = s(1,i);
            end
        end
    end
end
end
for i=1:c(1,1) % Cada renglón de la matriz es ocupado por el
    Q(i,:) = P(s(1,i),:); % cromosoma que indica el vector s.
end

function [d] = vec_des(M)
% Función:
% - vec_des es el nombre de la función que genera el vector de desviaciones estándar de los activos.
% Datos:

```

```

% - M es la matriz que en cada columna contiene la serie de rendimientos de un activo.
% Resultado:
% - d es el vector de desviaciones estándar.
c = size(M); % Se calculan las dimensiones de M.
d = zeros(1,c(1,2)); % Se crea un vector para ubicar las desviaciones estándar.
for i=1:c(1,2) % Para cada columna de M, se calcula la desviación estándar.
    d(1,i) = (var(M(:,i)))^(1/2);
end

```

```

function [r] = vec_ren(M)
% Función:
% - vec_ren es el nombre de la función que genera el vector de rendimientos promedio.
% Datos:
% - M es la matriz que en cada columna contiene la serie de rendimientos de un activo.
% Resultado:
% - r es el vector de rendimientos.
c = size(M); % Se calculan las dimensiones de M.
r = zeros(1,c(1,2)); % Se crea el vector para ubicar los rendimientos promedio.
for i=1:c(1,2)
    r(1,i) = mean(M(:,i)); % Se calcula el rendimiento promedio de cada activo.
end

```

Algoritmo genético 2

```

function [Q,c2,u3] = ag2_det(M,u,nc,ni,pc,pm)
% Función:
% - ag2_det es el nombre de la función que corre el algoritmo genético 2, determinación de los pesos
% de los activos financieros, utilizando como operadores genéticos el cruzamiento y la mutación.
% Datos:
% - M es la matriz que contiene las series históricas de los rendimientos de
% los activos por considerar, cada serie histórica ocupa una columna de la matriz.
% - u es el vector que indica los activos que integrarán la cartera de inversión.
% - nc es el número de cromosomas o individuos de la población.
% - ni es el número máximo de iteraciones.
% - pc es la probabilidad de cruzamiento.
% - pm es la probabilidad de mutación.
% Resultados:
% - Q es la matriz de la última generación en codificación decimal.
% - c2 es el cromosoma con mayor aptitud de la última generación en codificación decimal.
% - u3 es el valor de aptitud del cromosoma c2.
c = size(u); % Se calculan las dimensiones de u.
r = vec_ren2(M,u); % Se calcula el vector de rendimiento promedio, el vector de
d = vec_des2(M,u); % desviaciones estándar y la matriz de coeficientes de
C = mat_cor2(M,u); % correlación.
t = 0; % Se inicializa el contador de iteraciones.
P = pob_ini2(nc,c(1,2)); % Se genera la población inicial.
c1 = size(P); % Se calculan las dimensiones de P.
Q = con_dec2(P); % Se decodifica la población binaria en notación decimal.
while t < ni % Se inician las iteraciones del algoritmo genético.
    t = t+1; % Se actualiza el contador.
    e = zeros(1,c1(1,2)); % Se crea un vector para colocar al mejor cromosoma de la población.
    e1 = 0; % Se crea una variable falso / verdadero.
    v = eva_pob2(Q,r,d,C); % Se evalúa la población actual.
    u1 = max(v); % Se identifica la mejor aptitud de la población actual.
    for i=1:nc % Se ubica la posición de la mejor aptitud en la población actual.
        if v(1,i) == u1 % de acuerdo al vector de evaluaciones y se copia el cromosoma

```

```

    e(1,:)=P(i,:); % correspondiente de P en el vector e.
else
    c = e;
end
end
R = pob_sel2(P,v); % Se seleccionan los individuos de P que podrán seguir adelante.
T = ope_cru2(R,pc); % Se aplica el operador cruzamiento.
P = ope_mut2(T,pm); % Se aplica el operador mutación.
Q = con_dec2(P); % Se decodifica la población binaria en notación decimal.
v = eva_pob2(Q,r,d,C); % Se evalúa la población decimal.
u2 = min(v); % Se identifica la peor aptitud de la población modificada.
if u1>u2 % Se aplica elitismo, se compara la aptitud del peor cromosoma de
for i=1:nc % de la población modificada con la mejor aptitud de la
    if v(1,i)==u2 % población que inició la iteración. En caso de que la aptitud
        if e1==0 % del mejor cromosoma de la primera población sea mejor que la
            P(i,:)=e(1,:); % aptitud del peor cromosoma de la población modificada, el
                e1 = 1; % primero sustituye al segundo en la población modificada; en
                    else % caso contrario, la población modificada queda intacta.
                        P = P;
                    end
                end
            else
                P = P;
            end
        end
    end
else
    P = P;
end
end
Q = con_dec2(P); % Se decodifica la población binaria en notación decimal.
end
v = eva_pob2(Q,r,d,C); % Se evalúa la población.
u3 = max(v); % Se ubica la mejor aptitud obtenida en la corrida.
e2 = zeros(1,c(1,2)); % Se crea un vector para colocar al mejor cromosoma.
for i=1:nc % Se copia al mejor cromosoma.
    if v(1,i)==u3
        e2(1,:) = Q(i,:);
    else
        e2 = e2;
    end
end
end
e2 % Se muestra al mejor cromosoma encontrado en la corrida.
u3 % Se muestra el valor de aptitud del mejor cromosoma.

```

```
function [Q]=con_dec2(P)
```

```
% Función:
```

```
% - con_dec2 es la función que convierte a la matriz de la población binaria a una
```

```
% matriz de números decimales.
```

```
% Dato:
```

```
% - P es una matriz en la que cada renglón representa a un individuo de forma binaria.
```

```
% Resultado:
```

```
% - Q es una matriz en la que cada renglón contiene los pesos que deben ser asignados
```

```
% a cada activo que conforma la cartera de inversión.
```

```
c = size(P); % Se calculan las dimensiones de P.
```

```
Q = zeros(c(1,1),c(1,2)/14); % Se crea la matriz que contendrá cromosomas decimales.
```

```
a = zeros(c(1,2)/14,14); % Se crean dos matrices auxiliares para la conversión.
```

```
a1 = zeros(c(1,2)/14,1);
```

```
for i=1:c(1,1) % Para cada cromosoma de la población, se le divide en
```

```

for j=1:c(1,2)/14      % segmentos de 14 dígitos para colocarlos en la matriz a.
    for k=1:14
        a(j,k) = P(i,14*(j-1)+k);
    end
end
a1 = bi2de(a);        % Cada renglón de a se convierte a su equivalente decimal.
for j=1:c(1,2)/14    % Cada entrada de a1 es convertida a un número decimal entre
    Q(i,j) = a1(j,1)/((2^14)-1); % 0 y 1; su valor es colocado en la matriz Q en su lugar
end                  % correspondiente.
end
v = zeros(1,c(1,1)); % Se crea un vector auxiliar.
for i=1:c(1,1)       % A cada entrada se le asigna la suma de las entradas de cada
    v(1,i) = sum(Q(i,:)); % renglón de Q.
end
for i=1:c(1,1)       % Se normaliza cada renglón de la matriz Q.
    for j=1:c(1,2)/14
        Q(i,j) = Q(i,j)/v(1,i);
    end
end
end

```

```

function [c] = cot_ren2(r)
% Función:
% - cot_ren2 es el nombre de la función que calcula el mínimo rendimiento posible para una cartera que
% se puede obtener a partir de los rendimientos de los activos considerados para integrar la
% cartera de inversión.
% Dato:
% - r es el vector de rendimientos promedio de cada activo que integrará la cartera.
% Resultado:
% - c es el mínimo rendimiento posible.
a = sort(r); % Se ordenan los rendimientos.
c = a(1,1); % Se copia el peor rendimiento.
if c>0      % Se define el valor de la constante según el valor del peor rendimiento.
    c = 0;
else
    c = -c+0.0001;
end
end

```

```

function [v] = eva_pob2(P,r,d,C)
% Función:
% - eva_pob2 es el nombre de la función que genera el vector que contiene la
% evaluación de cada cromosoma de la población para el algoritmo genético 2.
% Datos:
% - P es la matriz que contiene en cada renglón a un individuo de la población; en esta
% matriz es el resultado de la conversión de la matriz binaria a una matriz decimal.
% - r es el vector de rendimientos promedio de cada activo que integrará la cartera.
% - d es el vector de desviaciones estándar de cada serie de rendimientos de los activos
% que integrarán la cartera.
% - C es la matriz de coeficientes de correlación existentes entre los activos que
% integrarán la cartera.
% Resultado:
% - v es el vector de evaluaciones de los cromosomas de la población.
a = size(P); % Se calculan las dimensiones de P.
v = zeros(1,a(1,1)); % Se crean el vector de aptitudes.
n = zeros(1,a(1,1)); % Se crean dos vectores auxiliares
m = zeros(1,a(1,1)); % para el cálculo de las aptitudes.
t = cot_ren2(r); % Se calcula la cota de rendimientos.
end

```

```

for i=1:a(1,1) % Se calcula el numerador de la
    n(1,i) = t; % aptitud de cada cromosoma.
end
for i=1:a(1,1)
    for j=1:a(1,2)
        n(1,i) = n(1,i)+P(i,j)*r(1,j);
    end
end
for i=1:a(1,1) % Se calcula el denominador de la
    for j=1:a(1,2) % aptitud de cada cromosoma.
        for k=1:a(1,2)
            m(1,i) = m(1,i)+P(i,j)*P(i,k)*C(j,k)*d(1,j)*d(1,k);
        end
    end
end
for i=1:a(1,1) % Se calcula el cociente que indica
    v(1,i) = n(1,i)/m(1,i); % la aptitud del cromosoma.
end

```

```
function [C] = mat_cor2(M,v)
```

```
% Función:
```

```
% - mat_cor2 es el nombre de la función que genera la matriz de coeficientes de
% correlación para los activos que conformarán la cartera de inversión.
```

```
% Datos:
```

```
% - M es la matriz que en cada columna contiene la serie de rendimientos de los activos
% disponibles para conformar la cartera de inversión.
```

```
% - v es el vector de activos que conforman la cartera de inversión.
```

```
% Resultado:
```

```
% - C es la matriz de coeficientes de correlación existentes entre los activos que
% conformarán la cartera de inversión.
```

```

d = size(v); % Se calculan las dimensiones de v.
C = zeros(d(1,2)); % Se crea la matriz que contendrá a los coeficientes.
N = zeros(2); % Se crea una matriz auxiliar.
for i=1:d(1,2) % Se calculan los coeficientes de correlación por
    for j=1:d(1,2) % por parejas y se rescata el elemento indicado
        N = corrccoef(M(:,v(1,i)),M(:,v(1,j))); % para integrar la matriz de coeficientes de
        C(i,j) = N(1,2); % correlación entre el total de activos.
    end
end

```

```
function [R] = ope_cru2(Q,pc)
```

```
% Función:
```

```
% - ope_cru2 es el nombre de la función que aplica el operador cruzamiento a la
% población generada para cruzamiento.
```

```
% Datos:
```

```
% - Q es la matriz que contiene en cada renglón a un individuo de la población en
% codificación binaria.
```

```
% - pc es la probabilidad de cruzamiento.
```

```
% Resultado
```

```
% - R es el matriz con la población creada después de la aplicación del operador
% cruzamiento en codificación binaria.
```

```

c = size(Q); % Se calculan las dimensiones de Q.
R = zeros(c); % Se crea la matriz resultado R.
r = rand(1,c(1,1)); % Se crea un vector de números aleatorios entre 0 y 1.
s = zeros(1,c(1,1)); % Se crea un vector auxiliar.
for i=1:c(1,1) % Se realizan las comparaciones

```

```

if r(1,i)<pc          % correspondientes para identificar
    s(1,i) = 1;      % los cromosomas por cruzar y se les
else                % señala en el vector auxiliar s.
    s(1,i) = s(1,i);
end
end
m = sum(s)/2;        % Se verifica que el número de
[m1,m2] = rat(m);    % cromosomas por cruzar sea par.
if m2==1            % En caso de no ser así, se genera
    s = s;          % un número aleatorio que defina
else                % el cromosoma que deberá ser
    r1 = randperm(c(1,1));
    if s(1,r1(1,1))==0
        s(1,r1(1,1)) = 1;
    else
        s(1,r1(1,1)) = 0;
    end
end
end
m3 = sum(s);        % Se calcula el número de cromosomas
if m3==0            % por cruzar. Si es 0, la población
    R = Q;          % queda intacta.
else                % Si el numero de cromosomas por
    R = Q;          % cruzar es mayor a 0, se crea un
    s1 = zeros(1,c(1,1));
    for i=1:c(1,1) % vector s1 donde cada entrada tiene
        s1(1,i) = i*s(1,i); % el número de renglón del cromosoma
    end            % por cruzar o un 0.
    U = zeros(m3,c(1,2)+1);
    k = 1;         % Se crea una matriz U de referencia.
    for i=1:c(1,1)
        if s1(1,i)==0
            U = U; % Se colocan en la primera columna
        else      % de U los números de renglones que
            U(k,1) = s1(1,i); % ocupan los cromosomas por cruzar
            k = k+1; % en la matriz original.
        end
    end
end
for i=1:m3          % Se copian los cromosomas que serán
    U(i,2:c(1,2)+1) = Q(U(i,1),:); % cruzados.
end
for i=1:2:m3-1     % Se aplica el cruzamiento.
    p = randperm(c(1,2)-1); % Se genera aleatoriamente el punto de cruce.
    R(U(i,1),p(1,1)+1:c(1,2)) = U(i+1,p(1,1)+2:c(1,2)+1); % Se intercambian los segmentos
    R(U(i+1,1),p(1,1)+1:c(1,2)) = U(i,p(1,1)+2:c(1,2)+1); % correspondientes
end                % para terminar el cruzamiento.
end
end

```

```
function [U] = ope_mut2(R,pm)
```

```
% Función:
```

```
% - ope_mut2 es el nombre de la función que aplica el operador mutación a la matriz
% que contiene a la población en codificación binaria.
```

```
% Datos:
```

```
% - R es la matriz que contiene en cada renglón a un individuo de la población en
% codificación binaria.
```

```
% - pm es la probabilidad de mutación.
```

```
% Resultado:
```

```

% - U es el matriz con la población después de la aplicación del operador mutación en
% codificación binaria.
c = size(R); % Se calculan las dimensiones de R.
U = zeros(c); % Se crea la matriz resultado U.
U = R; % Se copia la información de R en U.
T = rand(c); % Se crea una matriz de números aleatorios entre 0 y 1.
for i=1:c(1,1)*c(1,2) % Se realizan las comparaciones necesarias para definir si un gen
    if T(i)<pm % debe mutar o no y se ejecuta la mutación.
        if U(i)==0
            U(i) = 1;
        else
            U(i) = 0;
        end
    else
        U = U;
    end
end
end

```

```
function [P]=pob_ini2(nc,na)
```

```
% Función:
```

```
% - pob_ini2 es la función que genera la población inicial del algoritmo genético 2,
% cada individuo es una cadena de 14*na unos o ceros.
```

```
% Datos:
```

```
% - nc es el número de individuos en la población.
% - na es el número de activos que contiene la cartera.
```

```
% Resultado:
```

```
% - P es una matriz en la que cada renglón representa a un individuo.
```

```
P = randint(nc,14*na); % Se genera aleatoriamente la matriz en codificación binaria.
```

```
function [Q] = pob_sel2(P,v)
```

```
% Función:
```

```
% - pob_sel2 es el nombre de la función que genera una nueva población a partir de la población
% actual para que posteriormente a esta nueva generación le sean aplicados los operadores
% genéticos implementados en el algoritmo genético 2.
```

```
% Datos:
```

```
% - P es la matriz que contiene en cada renglón a un individuo de la población en codificación
% binaria.
```

```
% - v es el vector de evaluaciones.
```

```
% Resultado:
```

```
% - Q es la matriz formada por los individuos seleccionados de la generación actual para que les
% sean aplicados los operadores genéticos.
```

```
c = size(P); % Se calculan las dimensiones de P.
```

```
Q = zeros(c); % Se crea la matriz resultado Q.
```

```
f = sum(v); % Se calcula la suma total de aptitudes.
```

```
p = zeros(1,c(1,1)); % Se crea un vector auxiliar.
```

```
for i=1:c(1,1) % Se calcula el vector que mostrará la función de
    p(1,i) = v(1,i)/f; % de densidad de la población.
end
```

```
q = zeros(1,c(1,1)); % Se crea un vector auxiliar.
```

```
q(1,1) = p(1,1); % Se calcula el vector que mostrará la función de
for i=2:c(1,1) % distribución de la población.
```

```
    q(1,i) = q(1,i-1)+p(1,i);
end
```

```
r = rand(1,c(1,1)); % Se genera un vector de números aleatorios entre 0 y 1.
```

```
s = zeros(1,c(1,1)); % Se crea un vector auxiliar.
```

```
for i=1:c(1,1) % Se realizan las comparaciones necesarias para
```

```

if r(1,i)<q(1,1)           % identificar los cromosomas que serán
    s(1,i) = 1;           % seleccionados. En cada entrada del vector s, se
else                       % colocará el número de renglón en el que se ubica
    for j=1:c(1,1)-1     % el cromosoma seleccionado.
        if r(1,i)>q(1,j) & r(1,i)<q(1,j+1)
            s(1,i) = j+1;
        else
            s(1,i) = s(1,i);
        end
    end
end
end
end
for i=1:c(1,1)           % Se copian los cromosomas seleccionados en la matriz
    Q(i,:) = P(s(1,i,:)); % resultado Q.
end

```

```

function [r] = vec_des2(M,v)
% Función:
% - vec_des2 es el nombre de la función que genera el vector de desviaciones estándar
% de los activos que conformarán la cartera de inversión.
% Datos:
% - M es la matriz que en cada columna contiene la serie de rendimientos de de un activo.
% - v es el vector que contiene a los activos que integrarán la cartera de inversión.
% Resultado:
% - r es el vector de desviaciones estándar de los activos que integrarán la cartera de inversión.
a = size(v);           % Se calculan las dimensiones de v.
r = zeros(a);          % Se crea el vector de desviaciones estándar.
for i=1:a(1,2)         % Se calculan las desviaciones estándar.
    r(1,i) = (var(M(:,v(1,i))))^(1/2);
end

```

```

function [r] = vec_ren2(M,v)
% Función:
% - vec_ren2 es el nombre de la función que genera el vector de rendimientos de los
% activos que conformarán la cartera de inversión.
% Datos:
% - M es la matriz que en cada columna contiene la serie de rendimientos de un activo.
% - v es el vector que contiene a los activos que integrarán la cartera de inversión.
% Resultado:
% - r es el vector de rendimientos promedio de los activos que integrarán la cartera de inversión.
a = size(v);           % Se calculan las dimensiones de v.
r = zeros(a);          % Se crea el vector de rendimientos promedio.
for i=1:a(1,2)         % Se calculan los rendimientos promedio.
    r(1,i) = mean(M(:,v(1,i)));
end

```

Algoritmo genético 3

```

function [Q,e2,u3] = ag3_pmv(M,u,nc,ni,pc,pm)
% Función:
% - ag3_pmv es el nombre de la función que corre el algoritmo genético 3,
% determinación de los pesos de los activos financieros para crear la cartera de inversión de
% mínima varianza, utilizando como operadores genéticos el cruzamiento y la mutación.
% Datos:
% - M es la matriz que contiene las series históricas de los rendimientos de
% los activos por considerar, cada serie histórica ocupa una columna de la matriz.

```

```

% - u es el vector que indica los activos que integrarán la cartera de inversión.
% - nc es el número de cromosomas o individuos de la población.
% - ni es el número máximo de iteraciones.
% - pc es la probabilidad de cruzamiento.
% - pm es la probabilidad de mutación.
% Resultados
% - Q es la matriz de la última generación en codificación decimal.
% - e2 es el cromosoma con mayor aptitud de la última generación en codificación decimal.
% - u3 es el valor de aptitud del cromosoma e2.
c = size(u); % Se calculan las dimensiones de u, el vector de las
d = vec_des3(M,u); % desviaciones estándar y la matriz de coeficientes de
C = mat_cor3(M,u); % correlación.
t = 0; % Se inicializa el contador de iteraciones.
P = pob_ini3(nc,c(1,2)); % Se genera la población inicial.
cl = size(P); % Se calculan las dimensiones de P.
Q = con_dec3(P); % Se decodifica la población binaria en notación decimal.
while t<ni % Se inician las iteraciones del algoritmo genético.
    t = t+1; % Se actualiza el contador.
    e = zeros(1,c(1,2)); % Se crea un vector para colocar al mejor cromosoma de la población.
    e1 = 0; % Se crea una variable falso / verdadero.
    v = eva_pob3(Q,d,C); % Se evalúa la población actual.
    u1 = max(v); % Se identifica la mejor aptitud de la población actual.
    for i=1:nc % Se ubica la posición de la mejor aptitud en la población actual
        if v(1,i)==u1 % de acuerdo al vector de evaluaciones y se copia el cromosoma
            e(1,:) = P(i,:); % correspondiente de P en el vector e.
        else
            e = e;
        end
    end
    R = pob_sel3(P,v); % Se seleccionan los individuos de P que podrán seguir adelante.
    T = ope_cru3(R,pc); % Se aplica el operador cruzamiento.
    P = ope_mu3(T,pm); % Se aplica el operador mutación.
    Q = con_dec3(P); % Se decodifica la población binaria en notación decimal.
    v = eva_pob3(Q,d,C); % Se evalúa la población decimal.
    u2 = min(v); % Se identifica la peor aptitud de la población modificada.
    if u1>u2 % Se aplica elitismo, se compara la aptitud del peor cromosoma de
        for i=1:nc % de la población modificada con la mejor aptitud de la
            if v(1,i)==u2 % población que inicio la iteración. En caso de que la aptitud
                if e1==0 % del mejor cromosoma de la primera población sea mejor que la
                    P(i,:)=e(1,:); % aptitud del peor cromosoma de la población modificada, el
                    e1 = 1; % primero sustituye al segundo en la población modificada; en
                else % caso contrario, la población modificada queda intacta.
                    P = P;
                end
            end
        else
            P = P;
        end
    end
end
else
    P = P;
end
Q = con_dec3(P); % Se decodifica la población binaria en notación decimal.
end
v = eva_pob3(Q,d,C); % Se evalúa la población.
u3 = max(v); % Se ubica la mejor aptitud obtenida en la corrida.
e2 = zeros(1,c(1,2)); % Se crea un vector para colocar al mejor cromosoma.

```

```

for i=1:nc                % Se copia al mejor cromosoma.
    if v(1,i)==u3
        e2(1,:)= Q(i,:);
    else
        e2 = e2;
    end
end
e2                        % Se muestra al mejor cromosoma encontrado en la corrida.
u3                        % Se muestra el valor de aptitud del mejor cromosoma.

```

function [Q]=con_dec3(P) (Mismo código que function [Q]=con_dec2(P) del algoritmo genético 2).

```

function [c] = cot_des3(d)
% Función:
% - cot_des3 es el nombre de la función que calcula la cota de las desviaciones estándar
%   para una cartera que se puede obtener a partir de los activos considerados para
%   integrar la cartera de inversión.
% Datos:
% - d es el vector de desviaciones estándar de cada activo que integrará la cartera.
% Resultado:
% - c es la mayor desviación estándar más una constante.
a = size(d);              % Se calculan las dimensiones de d.
b = sort(d);              % Se ordenan las desviaciones estándar.
c = b(1,a(1,2))+0.0001;   % Se define el valor de la cota.

```

```

function [m] = eva_pob3(P,d,C)
% Función:
% - eva_pob3 es el nombre de la función que genera el vector que contiene la
%   evaluación de cada cromosoma de la población para el algoritmo genético 3.
% Datos:
% - P es la matriz que contiene en cada renglón a un individuo de la población; esta
%   matriz es el resultado de la conversión de la matriz binaria a una matriz decimal.
% - d es el vector de desviaciones estándar de cada serie de rendimientos de los activos
%   que integrarán la cartera.
% - C es la matriz de coeficientes de correlación existentes entre los activos que
%   integrarán la cartera
% Resultado
% - m es el vector de evaluaciones de los cromosomas de la población.
a = size(P);              % Se calculan las dimensiones de P.
m = zeros(1,a(1,1));     % Se crea el vector de aptitudes.
t = cot_des3(d);         % Se calcula la cota de desviaciones.
for i=1:a(1,1)           % Se calcula la aptitud del cromosoma.
    for j=1:a(1,2)
        for k=1:a(1,2)
            m(1,i) = m(1,i)+P(i,j)*P(i,k)*C(j,k)*d(1,j)*d(1,k);
        end
    end
end
end
for i=1:a(1,1)
    m(1,i) = t-m(1,i);
end

```

function [C] = mat_cor3(M,v) (Mismo código que function [C] = mat_cor2(M,v) del algoritmo genético 2).

function [R] = ope_cru3(Q,pc) (Mismo código que function [R] = ope_cru2(Q,pc) del algoritmo genético 2).

function [U] = ope_mut3(R,pm) (Mismo código que function [U] = ope_mut2(R,pm) del algoritmo genético 2).

function [P]=pob_ini3(nc,na) (Mismo código que function [P]=pob_ini2(nc,na) del algoritmo genético 2).

function [Q] = pob_sel3(P,v) (Mismo código que function [Q] = pob_sel2(P,v) del algoritmo genético 2).

function [r] = vec_des3(M,v) (Mismo código que function [r] = vec_des2(M,v) del algoritmo genético 2).

Algoritmo genético 4

function [Q,e2,u3] = ag4_ill(M,u,lr,nc,ni,pc,pm)

% Función:

% -ag4_ill es el nombre de la función que corre el algoritmo genético 4,
% determinación de los pesos de los activos financieros para formar una cartera óptima
% al considerar el rendimiento de un activo libre de riesgo, utilizando como operadores
% genéticos el cruzamiento y la mutación.

% Datos:

% -M es la matriz que contiene las series históricas de los rendimientos de
% los activos por considerar, cada serie histórica ocupa una columna de la matriz.
% -u es el vector que indica los activos que integrarán la cartera de inversión.
% -lr es el rendimiento del activo libre de riesgo.
% -nc es el número de cromosomas o individuos de la población.
% -ni es el número máximo de iteraciones.
% -pc es la probabilidad de cruzamiento.
% -pm es la probabilidad de mutación.

% Resultados:

% -Q es la matriz de la última generación en codificación decimal.
% -e2 es el cromosoma con mayor aptitud de la última generación en codificación decimal.
% -u3 es el valor de aptitud del cromosoma e2.

c = size(u); % Se calculan las dimensiones de u.
r = vec_ren4(M,u); % Se calcula el vector de rendimiento promedio, el vector de
d = vec_des4(M,u); % desviaciones estándar y la matriz de coeficientes de
C = mat_cor4(M,u); % correlación.
t = 0; % Se inicializa el contador de iteraciones.
P = pob_ini4(nc,c(1,2)); % Se genera la población inicial.
c1 = size(P); % Se calculan las dimensiones de P.
Q = con_dec4(P); % Se decodifica la población binaria en notación decimal.
while t<ni % Se inician las iteraciones del algoritmo genético.
t = t+1; % Se actualiza el contador.
e = zeros(1,c1(1,2)); % Se crea un vector para colocar al mejor cromosoma de la población.
e1 = 0; % Se crea una variable falso / verdadero.
v = eva_pob4(Q,r,d,C,lr); % Se evalúa la población actual.
u1 = max(v); % Se identifica la mejor aptitud de la población actual.
for i=1:nc % Se ubica la posición de la mejor aptitud en la población actual.
if v(1,i)==u1 % de acuerdo al vector de evaluaciones y se copia el cromosoma
e(1,:) = P(i,:); % correspondiente de P en el vector e.
else
e = e;
end
end
R = pob_sel4(P,v); % Se seleccionan los individuos de P que podrán seguir adelante.
T = ope_cru4(R,pc); % Se aplica el operador cruzamiento.
P = ope_mut4(T,pm); % Se aplica el operador mutación.
Q = con_dec4(P); % Se decodifica la población binaria en notación decimal.
v = eva_pob4(Q,r,d,C,lr); % Se evalúa la población decimal.

```

u2 = min(v); % Se identifica la peor aptitud de la población modificada.
if u1>u2 % Se aplica elitismo, se compara la aptitud del peor cromosoma de
    for i=1:nc % de la población modificada con la mejor aptitud de la
        if v(1,i)==u2 % población que inició la iteración. En caso de que la aptitud
            if e1==0 % del mejor cromosoma de la primera población sea mejor que la
                P(i,:)=e(1,:); % aptitud del peor cromosoma de la población modificada, el
                e1 = 1; % primero sustituye al segundo en la población modificada; en
            else % caso contrario, la población modificada queda intacta.
                P = P;
            end
        else
            P = P;
        end
    end
end
else
    P = P;
end
Q = con_dec4(P); % Se decodifica la población binaria en notación decimal.
end
v = eva_pob4(Q,r,d,C,lr); % Se evalúa la población.
u3 = max(v); % Se ubica la mejor aptitud obtenida en la corrida.
e2 = zeros(1,c(1,2)); % Se crea un vector para colocar al mejor cromosoma.
for i=1:nc % Se copia al mejor cromosoma.
    if v(1,i)==u3
        e2(1,:) = Q(i,:);
    else
        e2 = e2;
    end
end
e2 % Se muestra al mejor cromosoma encontrado en la corrida.
u3 % Se muestra el valor de aptitud del mejor cromosoma.

```

function [Q]=con_dec4(P) (Mismo código que function [Q]=con_dec2(P) del algoritmo genético 2).

```

function [v] = eva_pob4(P,r,d,C,lr)
% Función:
% - eva_pob4 es el nombre de la función que genera el vector que contiene la
% evaluación de cada cromosoma de la población para el algoritmo genético 4.
% Datos:
% - P es la matriz que contiene en cada renglón a un individuo de la población; esta
% matriz es el resultado de la conversión de la matriz binaria a una matriz decimal.
% - r es el vector de rendimientos promedio de cada activo que integrará la cartera.
% - d es el vector de desviaciones estándar de cada serie de rendimientos de los activos
% que integrarán la cartera.
% - C es la matriz de coeficientes de correlación existentes entre los activos que integrarán la cartera.
% - lr es el rendimiento del activo libre de riesgo.
% Resultado:
% - v es el vector de evaluaciones de los cromosomas de la población.
a = size(P); % Se calculan las dimensiones de P.
v = zeros(1,a(1,1)); % Se crean el vector de aptitudes.
n = zeros(1,a(1,1)); % Se crean dos vectores auxiliares
m = zeros(1,a(1,1)); % para el cálculo de las aptitudes.
for i=1:a(1,1) % Se calcula el numerador de la
    for j=1:a(1,2) % aptitud de cada cromosoma. En caso
        n(1,i) = n(1,i)+P(i,j)*r(1,j); % de ser negativo, se le asigna un
    end % valor positivo.

```

```

end
for i=1:a(1,1)
    n(1,i) = n(1,i)-lr;
end
for i=1:a(1,1)
    if n(1,i) > 0
        n(1,i) = n(1,i);
    else
        n(1,i) = 0.0001;
    end
end
for i=1:a(1,1)
    for j=1:a(1,2)
        for k=1:a(1,2)
            m(1,i) = m(1,i)+P(i,j)*P(i,k)*C(j,k)*d(1,j)*d(1,k);
        end
    end
end
for i=1:a(1,1)
    v(1,i) = (n(1,i)/(m(1,i))^(1/2));
end

```

% Se calcula el denominador de la
% aptitud de cada cromosoma.

% Se calcula el cociente que indica
% la aptitud del cromosoma.

function [C] = mat_cor4(M,v) (Mismo código que function [C] = mat_cor2(M,v) del algoritmo genético 2).

function [R] = ope_cru4(Q,pc) (Mismo código que function [R] = ope_cru2(Q,pc) del algoritmo genético 2).

function [U] = ope_mut4(R,pm) (Mismo código que function [U] = ope_mut2(R,pm) del algoritmo genético 2).

function [P]=pob_ini4(nc,na) (Mismo código que function [P]=pob_ini2(nc,na) del algoritmo genético 2).

function [Q] = pob_sel4(P,v) (Mismo código que function [Q] = pob_sel2(P,v) del algoritmo genético 2).

function [r] = vec_des4(M,v)

% Función:

% - vec_des4 es el nombre de la función que genera el vector de desviaciones estándar
% de los activos que conformarán la cartera de inversión.

% Datos

% - M es la matriz que en cada columna contiene la serie de rendimientos de un activo.

% - v es el vector que contiene a los activos que integrarán la cartera de inversión.

% Resultado:

% - r es el vector de desviaciones estándar de los activos que integrarán la cartera de inversión.

d = size(v); % Se calculan las dimensiones de v.

r = zeros(d); % Se crea el vector de desviaciones estándar.

for i=1:d(1,2) % Se calculan las desviaciones estándar.

r(1,i) = (var(M(:,v(1,i))))^(1/2);

end

function [r] = vec_ren4(M,v)

% Función:

% - vec_ren4 es el nombre de la función que genera el vector de rendimientos de los activos
% que conformarán la cartera de inversión.

% Datos:

% - M es la matriz que en cada columna contiene la serie de rendimientos de un activo.

% - v es el vector que contiene a los activos que integrarán la cartera de inversión.

% Resultado:

```

% - r es el vector de rendimientos de los activos que integrarán la cartera de inversión.
d = size(v); % Se calculan las dimensiones de v.
r = zeros(d); % Se crea el vector de rendimientos promedio.
for i=1:d(1,2) % Se calculan los rendimientos promedio.
    r(1,i) = mean(M(:,v(1,i)));
end

```

Algoritmo genético 5

```

function [Q,e2,u3] = ag5_iar(rp,dp,lr,a,nc,ni,pc,pm)
% Función:
% - ag5_iar es el nombre de la función que corre el algoritmo genético 5,
% determinación de la cartera óptima para el inversionista considerando su grado de
% aversión al riesgo, utilizando como operadores genéticos el cruzamiento y la mutación.
% Datos:
% - rp es el rendimiento de la cartera de mercado.
% - dp es la desviación estándar de la cartera de mercado.
% - lr es el rendimiento del activo libre de riesgo.
% - a es el grado de aversión al riesgo del inversionista.
% - nc es el número de cromosomas o individuos de la población.
% - ni es el número máximo de iteraciones.
% - pc es la probabilidad de cruzamiento.
% - pm es la probabilidad de mutación.
% Resultados:
% - Q es la matriz de la última generación en codificación decimal.
% - e2 es el cromosoma con mayor aptitud de la última generación en codificación decimal.
% - u3 es el valor de aptitud del cromosoma e2.
t = 0; % Se inicializa contador.
P = pob_ini5(nc); % Se genera población inicial.
c1 = size(P); % Se calculan las dimensiones de P.
Q = con_dec5(P); % Se decodifica la población binaria en notación decimal.
while t < ni % Se inician las iteraciones del algoritmo genético.
    t = t + 1; % Se actualiza el contador.
    e = zeros(1,c1(1,2)); % Se crea un vector para colocar al mejor cromosoma de la población.
    e1 = 0; % Se crea una variable falso / verdadero.
    v = eva_pob5(Q,rp,dp,lr,a); % Se evalúa la población actual.
    u1 = max(v); % Se identifica la mejor aptitud de la población actual.
    for i = 1:nc % Se ubica la posición de la mejor aptitud en la población actual
        if v(1,i) == u1 % de acuerdo al vector de evaluaciones y se copia el cromosoma
            e(1,:) = P(i,:); % correspondiente de P en el vector e.
        else
            e = e;
        end
    end
    R = pob_sel5(P,v); % Se seleccionan los individuos de P que podrán seguir adelante.
    T = ope_cru5(R,pc); % Se aplica el operador cruzamiento.
    P = ope_mut5(T,pm); % Se aplica el operador mutación.
    Q = con_dec5(P); % Se decodifica la población binaria en notación decimal.
    v = eva_pob5(Q,rp,dp,lr,a); % Se evalúa la población decimal.
    u2 = min(v); % Se identifica la peor aptitud de la población modificada.
    if u1 > u2 % Se aplica elitismo, se compara la aptitud del peor cromosoma de
        for i = 1:nc % de la población modificada con la mejor aptitud de la
            if v(1,i) == u2 % población que inició la iteración. En caso de que la aptitud
                if e1 == 0 % del mejor cromosoma de la primera población sea mejor que la
                    P(i,:) = e(1,:); % aptitud del peor cromosoma de la población modificada, el
                    e1 = 1; % primero sustituye al segundo en la población modificada; en
                end
            end
        end
    end
end

```

```

else
    P = P;
end
else
    P = P;
end
end
else
    P = P;
end
Q = con_dec5(P); % Se decodifica la población binaria en notación decimal.
end
v = eva_pob5(Q,rp,dp,lr,a); % Se evalúa la población.
u3 = max(v); % Se ubica la mejor aptitud obtenida en la corrida.
e2 = zeros(1,1); % Se crea una variable para colocar al mejor cromosoma.
for i=1:nc
    if v(1,i) == u3
        e2(1,:) = Q(i,:); % Se copia al mejor cromosoma.
    else
        e2 = e2;
    end
end
e2 % Se muestra al mejor cromosoma encontrado en la corrida.
u3 % Se muestra el valor de aptitud del mejor cromosoma.

function [Q]=con_dec5(P)
% Función:
% - con_dec5 es la función que convierte a la matriz de la población binaria a una matriz de
% números decimales.
% Dato:
% - P es una matriz en la que cada renglón representa a un individuo de forma binaria.
% Resultado:
% - Q es una matriz en la que cada renglón contiene el peso que deben ser asignados a la
% cartera de mercado para formar la cartera óptima del inversionista.
c = size(P); % Se calculan las dimensiones de P.
Q = zeros(c(1,1),c(1,2)/14); % Se crea la matriz que contendrá cromosomas decimales.
a = zeros(c(1,2)/14,14); % Se crean dos matrices auxiliares para la conversión.
a1 = zeros(c(1,2)/14,1);
for i=1:c(1,1) % Para cada cromosoma de la población, se le divide en
    for j=1:c(1,2)/14 % segmentos de 14 dígitos para colocarlos en la matriz a.
        for k=1:14
            a(j,k) = P(i,14*(j-1)+k);
        end
    end
end
a1 = bi2de(a); % Cada renglón de a se convierte a su equivalente decimal.
for j=1:c(1,2)/14 % Cada entrada de a1 es convertida a un numero decimal entre
    Q(i,j) = a1(j,1)/((2^14)-1); % 0 y 1; su valor es colocado en la matriz Q en su lugar
end % correspondiente.
end

function [v] = eva_pob5(P,rp,dp,lr,a)
% Función:
% - eva_pob5 es el nombre de la función que genera el vector que contiene la
% evaluación de cada cromosoma de la población para el algoritmo genético 5.
% Datos:
% - P es la matriz que contiene en cada renglón a un individuo de la población; esta

```

```

% matriz es el resultado de la conversión de la matriz binaria a una matriz decimal.
% - rp rendimiento de la cartera de mercado.
% - dp la desviación estándar de la cartera de mercado.
% - lr es el rendimiento del activo libre de riesgo.
% - a es el grado de aversión al riesgo del inversionista.
% Resultado:
% - v es el vector de evaluaciones de los cromosomas de la población.
a1 = size(P); % Se calculan las dimensiones de P.
v = zeros(1,a1(1,1)); % Se crea el vector de aptitudes.
for i=1:a1(1,1) % Se calcula la aptitud de cada
    v(1,i) = P(i,1)*rp+(1-P(i,1))*lr-(a*(P(i,1)^2)*(dp^2)/2); % cromosoma
end
v1 = sort(v); % Se crea un vector ordenado de aptitudes.
if v1(1,1)>0 % Se verifica que no existan aptitudes
    v = v; % negativas. En caso de haberlas,
% se ubican y se les asigna un valor
% positivo pequeño.
else
    for i=1:a1(1,1)
        if v(1,i) > 0
            v(1,i) = v(1,i);
        else
            v(1,i) = lr/1000;
        end
    end
end
end

function [R] = ope_cru5(Q,pc) (Mismo código que function [R] = ope_cru2(Q,pc) del algoritmo genético 2).

function [U] = ope_mut5(R,pm) (Mismo código que function [U] = ope_mut2(R,pm) del algoritmo genético 2).

function [P]=pob_ini5(nc)
% Función:
% - pob_ini5 es la función que genera la población inicial del algoritmo genético 5, cada
% individuo es una cadena de 14.
% Datos:
% - nc es el número de individuos en la población.
% Resultado:
% - P es una matriz en la que cada renglón representa a un individuo.
P = randint(nc,14); % Se genera aleatoriamente la matriz en codificación binaria.

function [Q] = pob_sel5(P,v) (Mismo código que function [Q] = pob_sel2(P,v) del algoritmo genético 2).

```

APÉNDICE 2: TABLAS DE RESULTADOS (PARÁMETROS GENÉTICOS)

Algoritmo genético 1: ag1_ele(M,15,nc,ni,pc,pm)

ag1_ele(M,15,nc,100,0.25,0.01)

nc	Cromosoma	Aptitud	Tiempo (s)
220	6,11,15,18,22,31,33,37,39,40,42,43,49,51,52	106.5630	81.109
360	6,8,11,15,18,30,31,39,40,41,42,43,49,51,52	106.5236	167.750
190	6,11,15,18,22,30,31,33,37,39,40,42,43,49,52	106.5224	65.953
440	6,8,11,15,18,26,29,31,37,39,40,42,43,49,52	106.4099	229.031
350	6,8,11,15,25,29,31,33,37,39,40,42,43,49,52	106.3389	160.484
380	6,11,15,18,22,29,31,37,39,41,42,43,49,51,52	106.2365	182.078
310	6,11,15,18,22,29,34,37,39,40,41,42,43,49,52	106.2357	133.469
370	6,11,15,18,22,25,30,33,37,39,40,42,43,49,52	106.1622	175.203
320	6,8,11,15,18,30,39,40,41,42,43,46,49,51,52	106.0915	140.063
90	8,11,15,18,29,34,37,39,40,42,43,46,49,51,52	106.0692	25.437
180	6,8,15,18,21,31,33,37,39,40,42,43,49,51,52	106.0671	61.391
20	6,11,18,24,29,30,31,33,37,39,40,42,43,49,52	106.0348	5.250
80	6,8,15,18,21,30,31,33,37,39,40,42,43,49,52	106.0191	22.016
270	5,6,11,15,18,22,25,39,40,41,42,43,49,51,52	105.9381	108.750
420	6,8,11,15,29,31,37,39,40,42,43,44,45,49,52	105.8786	213.062
150	6,11,15,18,29,31,33,34,37,38,39,40,42,43,52	105.8531	48.094
230	8,11,15,18,21,25,30,33,39,40,42,43,49,51,52	105.7876	86.157
340	6,8,15,18,24,29,33,37,38,39,40,43,49,51,52	105.7869	153.828
400	8,11,18,22,25,29,33,34,37,38,39,40,42,49,52	105.7490	197.407
450	11,15,17,18,21,22,25,29,31,37,39,40,42,49,52	105.7322	237.407
60	5,6,11,15,18,30,31,33,37,39,40,41,42,43,49	105.6695	15.875
140	5,6,8,15,18,25,30,31,37,39,40,42,43,49,52	105.6664	43.953
460	11,15,18,21,22,29,31,33,37,39,40,41,42,51,52	105.6300	246.797
390	5,6,8,15,18,29,30,31,37,38,39,40,42,49,52	105.6207	189.969
300	6,8,11,13,15,18,21,30,37,39,40,42,49,51,52	105.6127	127.672
70	6,8,11,13,15,18,29,32,34,37,39,40,49,51,52	105.5692	18.937
240	8,11,15,18,21,25,31,39,40,42,43,44,45,49,52	105.5067	91.687
110	5,6,8,11,18,25,29,37,39,40,41,42,49,51,52	105.4810	32.281
50	5,11,15,18,25,29,32,38,39,40,41,43,49,51,52	105.4422	13.015
280	6,18,20,25,29,32,33,34,37,39,40,42,43,49,52	105.4374	114.688
430	5,8,15,18,27,29,34,37,39,40,41,42,49,51,52	105.4327	221.110
480	8,11,13,15,18,25,29,33,39,40,43,49,50,51,52	105.3916	265.391
410	6,8,11,15,18,24,33,36,37,39,40,43,49,50,52	105.3864	205.125
490	1,6,8,9,15,18,29,33,37,39,40,42,43,49,51	105.3863	273.313
170	5,11,14,15,18,21,22,29,35,39,40,42,43,49,52	105.3860	56.968
210	5,8,11,18,21,22,25,27,31,39,40,42,43,49,52	105.3757	76.218
500	6,8,15,18,25,29,34,36,37,39,41,42,49,51,52	105.3217	281.500
260	11,18,25,29,33,34,36,37,39,40,42,43,44,51,52	105.2955	103.047
160	11,15,18,20,21,25,29,30,31,39,40,42,43,44,52	105.2750	52.469
10	7,8,15,18,25,29,32,33,34,39,40,42,43,49,52	105.2473	2.922
30	6,11,13,15,22,25,29,32,37,39,40,41,42,43,52	105.2361	7.672
130	8,11,21,24,25,29,30,33,37,39,40,42,43,49,52	105.2040	39.937
120	5,11,13,18,24,29,32,33,37,39,40,42,43,49,51	105.1291	36.110

330	6,15,18,22,24,25,32,33,39,40,42,43,44,49,52	105.1263	146.421
470	5,6,8,11,15,18,33,39,40,42,43,44,48,51,52	105.0624	256.563
200	6,15,18,21,31,33,34,36,37,39,40,43,46,49,51	104.9703	70.735
250	6,8,16,18,22,24,36,37,39,40,42,43,49,51,52	104.7175	97.157
290	5,6,11,14,18,30,31,33,39,40,43,48,49,51,52	104.5830	120.922
40	6,8,11,13,25,27,29,36,37,39,40,42,43,49,52	104.5056	10.421
100	11,13,15,20,21,22,29,31,37,39,40,42,46,49,51	104.3945	28.813

agl_ele(M,15,100,ni,0.25,0.01)

ni	Cromosoma	Aptitud	Tiempo (s)
310	6,8,11,15,18,22,29,31,37,39,40,42,43,49,52	106.7762	87.657
450	6,11,15,18,22,29,31,33,37,39,40,42,43,49,52	106.7567	127.015
480	6,11,15,18,22,29,31,37,39,40,42,43,49,51,52	106.7526	134.891
460	6,11,15,18,29,31,37,39,40,41,42,43,49,51,52	106.7225	129.532
360	6,8,11,15,18,29,31,37,39,40,42,43,49,51,52	106.6555	101.922
440	6,8,11,15,18,29,31,37,39,40,42,43,49,51,52	106.6555	123.844
470	6,8,11,15,18,29,31,37,39,40,42,43,49,51,52	106.6555	132.375
430	6,8,11,15,18,29,31,39,40,41,42,43,49,51,52	106.6519	121.188
420	6,11,15,18,30,31,33,37,39,40,42,43,49,51,52	106.6429	118.453
220	8,11,15,18,29,31,37,39,40,41,42,43,49,51,52	106.6266	62.593
400	6,8,11,15,18,22,29,37,39,40,41,42,43,49,52	106.6025	113.062
260	6,11,15,18,29,31,33,37,39,40,42,43,49,51,52	106.6019	73.718
130	6,11,15,18,29,30,31,37,39,40,42,43,49,51,52	106.5734	37.282
270	6,11,15,18,25,29,31,33,39,40,41,42,43,49,52	106.5168	76.344
230	6,8,11,15,18,22,31,37,39,40,41,42,43,49,52	106.5131	65.172
320	5,6,8,11,15,18,29,39,40,41,42,43,49,51,52	106.4853	90.266
370	5,6,8,11,15,18,29,31,39,40,41,42,43,49,52	106.4652	104.344
380	6,11,18,30,31,33,37,39,40,41,42,43,49,51,52	106.4625	107.281
490	6,11,18,30,31,33,37,39,40,41,42,43,49,51,52	106.4625	137.625
390	6,8,11,15,18,27,31,37,39,40,41,42,43,49,52	106.4510	109.844
330	8,11,15,18,22,29,31,34,37,39,40,42,49,51,52	106.4253	93.204
350	8,11,13,15,18,22,29,39,40,41,42,43,49,51,52	106.3635	98.796
210	8,11,15,18,25,29,31,33,39,40,41,43,49,51,52	106.3357	59.468
180	6,8,11,15,18,31,36,37,39,40,41,42,43,49,52	106.3253	51.391
340	6,8,11,15,18,31,33,36,37,39,40,43,49,51,52	106.3046	96.062
150	6,11,15,18,29,30,31,33,39,40,42,43,46,49,52	106.2519	43.063
410	6,11,18,25,29,30,31,33,39,40,42,43,49,51,52	106.2302	115.531
240	6,11,15,18,22,25,29,33,34,39,40,42,43,49,52	106.2099	67.953
250	5,6,8,11,15,18,21,27,31,39,40,42,43,49,52	106.1954	70.734
290	8,15,18,21,22,25,29,31,37,39,40,42,43,49,52	106.1891	82.094
280	6,15,18,25,30,31,33,36,37,39,40,42,43,49,52	106.1629	79.234
190	5,6,8,11,15,18,22,33,39,40,41,42,43,49,52	106.1346	54.156
500	11,15,18,22,24,25,29,32,39,40,41,42,43,49,52	106.0823	140.907
170	6,8,11,13,15,18,21,22,31,39,40,42,43,49,52	106.0404	48.422
300	6,8,11,15,18,21,22,37,39,40,41,42,43,49,52	106.0151	84.828
160	8,11,15,18,29,31,36,37,38,39,40,42,49,51,52	106.0133	45.687
120	6,11,15,18,20,21,25,29,31,37,39,40,42,49,52	106.0092	34.578
200	6,8,10,11,15,18,29,33,39,40,41,43,49,51,52	105.8898	56.781

90	11,15,18,22,25,29,30,31,34,37,39,40,42,43,52	105.8335	26.125
80	11,18,22,24,25,29,32,37,39,40,41,43,49,51,52	105.6028	23.219
30	5,8,11,15,18,22,25,34,37,39,40,42,49,51,52	105.4913	9.110
140	8,15,18,21,24,29,32,33,36,39,40,43,49,51,52	105.3391	40.047
110	6,8,11,13,15,18,22,29,34,39,40,43,44,51,52	105.3386	31.672
100	6,11,18,20,22,27,29,31,37,39,40,41,43,49,51	105.2103	28.797
50	6,8,11,18,25,29,31,33,34,37,38,39,43,49,51	105.0560	14.750
40	6,11,15,18,22,24,28,29,30,40,42,43,44,49,52	104.8750	12.016
60	15,18,22,30,32,34,36,37,39,40,42,49,50,51,52	104.8133	17.516
70	1,11,15,18,25,29,30,36,40,42,43,44,46,49,52	104.3189	20.328
10	1,5,11,15,18,20,21,22,25,29,37,39,49,51,52	104.0794	3.500
20	12,15,21,25,28,29,32,33,34,35,39,42,43,49,52	102.8267	6.296

ag1_ele(M,15,100,100,pc,0.01)

pc	Cromosoma	Aptitud	Tiempo (s)
0.39	11,15,18,20,22,24,29,31,37,39,40,42,43,49,52	106.3826	29.860
0.02	8,11,15,18,22,25,29,33,39,40,41,42,43,49,52	106.3772	26.609
0.31	6,8,11,15,18,22,29,33,37,39,40,42,49,51,52	106.2875	29.188
0.33	6,8,11,18,25,29,31,33,37,39,40,43,49,51,52	106.2771	29.328
0.29	8,11,15,18,24,25,29,31,39,40,41,42,43,49,52	106.2739	29.094
0.25	11,18,25,29,30,31,33,37,39,40,42,43,49,51,52	106.2557	28.703
0.23	5,8,11,15,18,24,29,33,39,40,41,43,49,51,52	106.2462	28.515
0.01	1,8,11,15,18,29,31,37,39,40,41,42,43,49,52	106.0914	26.500
0.05	6,8,11,15,18,21,29,31,37,39,41,42,49,51,52	105.9640	26.984
0.27	6,11,13,15,18,20,22,29,39,40,41,42,43,49,52	105.9633	28.984
0.15	8,11,15,18,22,29,30,31,33,39,40,42,49,51,52	105.9578	27.922
0.16	5,11,15,18,22,25,29,31,32,39,40,42,43,49,52	105.9399	28.062
0.44	6,8,11,18,25,29,33,34,37,39,40,42,43,49,52	105.8992	30.406
0.04	6,8,11,18,25,29,33,36,37,39,40,41,49,51,52	105.8311	26.859
0.37	5,8,11,15,18,22,29,31,38,39,40,43,49,51,52	105.7665	29.875
0.07	6,8,11,15,18,21,22,27,29,33,39,40,43,49,52	105.7034	27.188
0.36	11,15,21,25,29,30,33,37,39,40,42,43,49,51,52	105.6919	29.719
0.28	6,8,15,18,24,25,31,33,39,40,42,43,49,51,52	105.6531	29.062
0.18	11,15,18,20,22,24,29,37,39,40,43,49,50,51,52	105.6389	28.109
0.13	8,11,18,25,27,33,37,39,40,41,42,43,49,51,52	105.6119	27.704
0.22	11,15,18,25,29,31,33,34,36,37,39,40,43,51,52	105.5985	28.484
0.32	5,6,11,15,18,22,29,33,37,39,40,42,45,49,52	105.5829	29.281
0.42	11,15,18,21,22,24,29,31,36,37,39,40,43,51,52	105.5810	30.125
0.5	8,10,15,18,29,34,36,37,39,40,41,42,49,51,52	105.5701	30.937
0.17	11,18,21,22,24,25,31,37,39,40,42,43,49,51,52	105.5435	28.109
0.49	6,8,11,13,15,18,21,29,36,37,39,40,43,49,51	105.5301	30.938
0.38	5,6,8,15,18,21,22,29,31,36,37,39,40,49,52	105.4940	29.828
0.45	6,11,15,18,20,21,22,29,31,37,39,40,41,42,49	105.4750	30.469
0.21	5,11,13,15,18,29,31,33,37,39,40,42,43,49,51	105.4615	28.375
0.09	2,6,11,15,18,24,29,32,37,39,40,42,49,51,52	105.4414	27.328
0.34	5,8,11,15,18,24,30,31,39,40,41,44,49,51,52	105.4229	29.500
0.48	6,11,15,18,25,29,30,31,33,34,40,43,49,51,52	105.4011	30.703
0.43	6,11,15,18,24,25,30,32,33,37,40,42,43,49,52	105.3316	30.328

0.1	11,15,18,22,29,31,32,37,38,39,40,42,49,52,54	105.3071	27.360
0.12	15,18,21,22,24,29,31,33,39,40,41,43,49,51,52	105.2948	27.656
0.24	6,8,11,15,18,22,24,29,31,38,39,40,41,43,52	105.2863	28.656
0.41	1,6,11,13,18,21,29,31,33,37,39,40,43,49,51	105.2821	30.109
0.2	1,8,11,15,18,25,29,31,34,36,39,40,42,49,52	105.2784	28.313
0.19	6,11,15,18,21,25,29,31,35,37,39,42,43,49,52	105.2731	28.250
0.14	9,15,18,22,25,29,36,37,39,40,41,42,49,51,52	105.2568	27.844
0.4	6,15,18,24,25,31,32,33,37,39,40,41,43,51,52	105.2448	29.984
0.06	6,8,15,18,21,24,31,34,36,37,39,40,43,49,51	105.1876	27.094
0.03	6,8,15,18,21,22,25,29,34,37,40,43,49,51,52	105.1542	26.781
0.46	6,8,11,15,18,30,31,32,39,41,42,44,49,51,52	105.0659	30.594
0.3	5,8,11,15,18,22,30,31,32,37,39,40,42,43,49	104.8930	29.235
0.26	1,8,18,21,24,25,29,33,39,40,43,44,49,51,52	104.7819	28.859
0.35	5,8,11,15,18,23,28,29,37,38,39,40,42,43,52	104.6199	29.594
0.47	11,14,15,18,22,24,29,31,32,35,37,39,40,42,52	104.4649	30.703
0.08	6,7,11,18,22,29,31,37,39,40,43,45,49,50,52	104.2787	27.235
0.11	5,6,8,9,17,29,30,33,37,39,40,41,43,49,51	104.2063	27.515

agl_ele(M,15,100,100,0.25,pm)

pm	Cromosoma	Aptitud	Tiempo (s)
0.095	6,8,11,15,29,31,33,37,39,40,42,43,49,51,52	106.3266	30.266
0.19	6,8,11,15,18,29,36,37,39,40,41,42,43,49,52	106.2928	30.547
0.16	6,8,11,15,18,29,38,39,40,41,42,43,49,51,52	106.1957	30.500
0.225	11,15,22,25,29,30,31,33,37,39,40,42,43,49,52	106.1848	30.594
0.165	4,8,11,15,18,25,29,31,37,39,40,42,43,49,52	106.1809	30.547
0.17	8,11,15,18,21,29,37,39,40,42,43,49,51,52,54	106.1411	30.375
0.23	11,15,18,22,29,30,31,36,37,39,40,42,43,49,52	106.1370	30.656
0.02	6,8,15,18,21,29,31,37,39,40,42,43,49,51,52	106.1012	29.500
0.025	11,15,18,22,25,29,31,34,37,39,40,42,43,51,52	106.1009	29.625
0.115	6,11,18,22,25,29,31,33,36,37,39,40,49,51,52	106.0541	30.266
0.135	6,8,15,18,29,32,37,39,40,41,42,43,49,51,52	106.0339	30.453
0.21	8,11,15,18,24,29,30,31,39,40,42,43,49,51,52	106.0184	30.672
0.105	6,8,15,18,25,29,31,33,36,37,39,40,41,49,52	105.9904	30.438
0.195	6,15,18,22,25,29,32,34,37,39,40,42,43,49,52	105.9408	30.641
0.015	6,8,11,18,29,30,31,33,37,39,42,43,49,51,52	105.9160	29.203
0.09	8,11,15,18,21,25,28,29,33,39,40,42,43,49,52	105.8566	30.219
0.05	6,8,11,15,18,22,25,29,37,39,40,41,42,43,49	105.8561	30.016
0.065	6,11,15,18,21,31,33,34,37,39,40,42,43,51,52	105.8237	30.219
0.07	8,11,15,18,25,29,30,34,39,40,42,43,49,51,52	105.8168	30.265
0.08	6,8,11,17,18,22,29,31,37,39,40,42,43,49,52	105.7713	30.343
0.24	6,8,18,21,22,29,31,33,37,39,40,42,43,49,51	105.6019	30.734
0.145	8,10,11,15,18,25,29,32,33,39,40,42,43,49,52	105.5761	30.391
0.005	5,6,8,18,25,30,33,36,37,39,40,42,43,49,52	105.5688	28.328
0.085	11,15,18,21,24,25,29,31,33,36,37,39,40,49,52	105.5638	30.187
0.18	6,11,15,18,22,24,29,32,39,40,43,44,49,51,52	105.5434	30.594
0.175	8,11,15,18,24,29,34,36,38,39,40,43,49,51,52	105.5302	30.703
0.04	6,8,11,15,18,21,30,31,37,39,40,43,45,49,52	105.4963	29.985
0.03	6,8,11,15,18,30,33,38,39,40,42,43,46,49,52	105.4958	29.734

0.035	11,15,18,20,25,29,30,33,37,39,40,41,42,49,52	105.4942	29.953
0.235	5,6,8,15,18,22,29,33,37,39,40,41,43,49,51	105.4399	30.671
0.055	1,6,11,18,24,25,33,37,39,40,41,42,43,49,52	105.4198	30.172
0.12	6,8,15,18,21,27,29,33,37,38,39,40,43,49,52	105.4083	30.453
0.2	6,8,11,15,18,21,22,29,34,37,39,42,49,52,54	105.3665	30.594
0.215	6,8,11,15,18,27,29,31,36,37,39,41,42,43,52	105.2402	30.671
0.245	6,15,18,22,24,25,29,31,32,37,39,41,42,49,52	105.1698	30.766
0.205	11,13,15,18,27,29,30,31,37,40,41,42,43,49,52	105.1667	30.688
0.13	15,18,21,25,27,29,31,39,40,42,43,46,49,51,52	105.0824	30.359
0.25	7,8,13,15,18,24,29,31,36,37,39,40,41,49,52	105.0366	30.813
0.06	6,11,18,21,22,24,29,31,37,38,39,49,50,51,52	105.0278	30.328
0.075	8,11,15,20,21,22,25,29,39,40,42,43,49,51,52	105.0260	30.344
0.11	9,15,18,21,29,30,33,36,37,39,40,43,49,51,52	105.0144	30.344
0.01	11,13,15,18,21,22,25,29,33,39,40,42,43,46,49	104.9953	28.938
0.15	5,8,15,18,21,22,26,29,31,39,40,43,49,51,52	104.9315	30.500
0.155	6,11,15,18,22,29,31,33,39,42,43,44,50,51,52	104.8384	30.454
0.125	6,18,21,24,25,31,33,34,37,38,39,40,42,49,52	104.7615	30.375
0.1	6,8,18,22,24,33,34,37,39,40,41,42,43,49,51	104.7375	30.391
0.185	8,15,18,21,29,30,32,34,36,37,39,40,42,43,49	104.7151	30.640
0.22	6,15,18,22,24,30,31,33,34,37,39,42,44,49,52	104.6221	30.718
0.14	15,18,20,21,22,24,29,33,37,39,43,49,51,52,53	104.3941	30.454
0.045	6,9,11,15,22,30,37,38,39,40,41,42,43,49,52	104.2448	30.000

Algoritmo genético 2: ag2_det(M,u,nc,ni,pc,pm)

ag2_det(M,u,nc,100,0.25,0.01)

nc	Cromosoma: u = (3,7,9,16,17,27,30,34,46,48)	Aptitud	Tiempo (s)
230	0.1689,0.0816,0.0801,0.0685,0.0111,0.2071,0.0556,0.1971,0.1300,0.0000	19.5509	135.593
420	0.1727,0.0758,0.0789,0.0651,0.0119,0.2216,0.0506,0.2000,0.1234,0.0000	19.5494	304.172
410	0.1763,0.0737,0.0693,0.0688,0.0086,0.2053,0.0620,0.1906,0.1454,0.0000	19.5493	293.937
320	0.1638,0.0704,0.0884,0.0665,0.0147,0.2159,0.0555,0.2013,0.1235,0.0000	19.5476	209.046
470	0.1623,0.0787,0.0790,0.0591,0.0173,0.2091,0.0610,0.1998,0.1333,0.0004	19.5463	358.313
80	0.1787,0.0745,0.0792,0.0687,0.0127,0.2191,0.0606,0.1965,0.1095,0.0005	19.5388	38.500
110	0.1848,0.0637,0.0792,0.0616,0.0024,0.2137,0.0666,0.1982,0.1297,0.0001	19.5382	55.266
440	0.1637,0.0864,0.0613,0.0648,0.0289,0.2179,0.0521,0.1990,0.1258,0.0001	19.5369	325.485
490	0.1841,0.0744,0.0582,0.0601,0.0146,0.2150,0.0571,0.1924,0.1435,0.0006	19.5360	380.562
430	0.1694,0.0785,0.0933,0.0626,0.0249,0.1909,0.0588,0.1874,0.1342,0.0000	19.5340	315.375
500	0.1726,0.0648,0.0767,0.0737,0.0319,0.1936,0.0605,0.1963,0.1299,0.0000	19.5336	392.281
50	0.1662,0.0645,0.0796,0.0559,0.0224,0.2128,0.0592,0.1946,0.1435,0.0013	19.5315	22.954
210	0.1742,0.0590,0.0728,0.0569,0.0162,0.2361,0.0597,0.1894,0.1355,0.0002	19.5298	120.766
400	0.1592,0.0712,0.0621,0.0590,0.0388,0.2205,0.0594,0.1978,0.1317,0.0003	19.5287	283.953
100	0.1750,0.0761,0.0994,0.0649,0.0141,0.1951,0.0619,0.1915,0.1219,0.0001	19.5278	49.641
340	0.1710,0.0756,0.0916,0.0552,0.0065,0.2180,0.0627,0.1792,0.1401,0.0001	19.5263	226.922
360	0.1906,0.0695,0.0630,0.0739,0.0197,0.2047,0.0656,0.1932,0.1192,0.0006	19.5241	245.234
240	0.1804,0.0621,0.0786,0.0545,0.0064,0.2187,0.0580,0.1916,0.1488,0.0009	19.5208	143.281
170	0.1820,0.0731,0.0650,0.0664,0.0278,0.1910,0.0593,0.1874,0.1472,0.0008	19.5186	92.891
460	0.1652,0.0671,0.0640,0.0659,0.0293,0.2021,0.0646,0.1950,0.1449,0.0019	19.5175	346.625
220	0.1540,0.0623,0.1061,0.0660,0.0197,0.2051,0.0637,0.1892,0.1339,0.0000	19.5163	127.985

290	0.1685,0.0845,0.0720,0.0692,0.0214,0.1821,0.0643,0.1890,0.1489,0.0001	19.5154	183.235
140	0.1722,0.0731,0.0971,0.0661,0.0178,0.1864,0.0572,0.1863,0.1431,0.0007	19.5145	73.469
70	0.1661,0.0697,0.0812,0.0718,0.0235,0.2121,0.0532,0.2041,0.1166,0.0017	19.5137	33.172
280	0.1719,0.0729,0.0892,0.0787,0.0303,0.1972,0.0564,0.1857,0.1176,0.0001	19.5114	174.735
380	0.1504,0.0764,0.1032,0.0622,0.0178,0.1945,0.0587,0.1877,0.1487,0.0004	19.5098	264.188
130	0.1668,0.0537,0.1076,0.0683,0.0033,0.2111,0.0616,0.1870,0.1405,0.0001	19.5089	67.437
150	0.1700,0.0795,0.0540,0.0576,0.0162,0.2228,0.0613,0.1988,0.1370,0.0028	19.5083	79.875
350	0.1559,0.0717,0.0550,0.0646,0.0289,0.2204,0.0567,0.2068,0.1383,0.0017	19.5075	235.657
300	0.1400,0.0718,0.0919,0.0764,0.0211,0.2170,0.0591,0.1898,0.1328,0.0001	19.5040	191.766
480	0.1750,0.0686,0.0912,0.0631,0.0123,0.2198,0.0581,0.2012,0.1085,0.0022	19.5020	369.297
390	0.1655,0.0793,0.0810,0.0828,0.0102,0.2064,0.0560,0.1793,0.1395,0.0000	19.5012	274.437
40	0.1790,0.0878,0.0617,0.0733,0.0030,0.2088,0.0587,0.1943,0.1313,0.0021	19.5010	18.110
160	0.1787,0.0645,0.0716,0.0710,0.0231,0.1956,0.0534,0.2160,0.1257,0.0004	19.4965	86.218
250	0.1710,0.0784,0.1034,0.0621,0.0063,0.1937,0.0667,0.1789,0.1386,0.0009	19.4960	150.688
200	0.1641,0.0692,0.0632,0.0830,0.0308,0.2123,0.0624,0.1866,0.1271,0.0013	19.4945	113.843
120	0.1866,0.0879,0.0781,0.0575,0.0075,0.2155,0.0542,0.2030,0.1078,0.0019	19.4852	61.203
90	0.1789,0.0727,0.1011,0.0649,0.0322,0.1910,0.0467,0.1939,0.1181,0.0005	19.4795	44.031
450	0.1679,0.0637,0.0634,0.0508,0.0300,0.2035,0.0633,0.2005,0.1539,0.0030	19.4727	335.687
10	0.1764,0.1040,0.0851,0.0631,0.0151,0.1874,0.0574,0.1873,0.1235,0.0007	19.4707	4.390
60	0.1665,0.0794,0.0942,0.0713,0.0017,0.1829,0.0653,0.1841,0.1511,0.0035	19.4528	28.578
180	0.1676,0.0832,0.0455,0.0596,0.0040,0.2307,0.0724,0.2119,0.1241,0.0010	19.4515	99.515
260	0.1770,0.0585,0.0623,0.0887,0.0190,0.2248,0.0439,0.2144,0.1110,0.0004	19.4334	159.000
190	0.1458,0.0765,0.0819,0.0831,0.0537,0.1926,0.0483,0.1915,0.1265,0.0001	19.4316	106.469
270	0.1392,0.0659,0.0591,0.0652,0.0324,0.2086,0.0737,0.1993,0.1544,0.0022	19.4291	166.906
30	0.1856,0.0879,0.0714,0.0893,0.0145,0.2131,0.0620,0.1813,0.0936,0.0013	19.4202	13.391
310	0.1667,0.0789,0.0736,0.0543,0.0550,0.1906,0.0331,0.2040,0.1435,0.0003	19.4003	200.171
370	0.1790,0.0531,0.0610,0.0540,0.0671,0.1808,0.0645,0.2060,0.1344,0.0001	19.3736	254.985
20	0.1367,0.0929,0.1200,0.0543,0.0201,0.2343,0.0639,0.1781,0.0986,0.0011	19.3440	8.844
330	0.1752,0.0764,0.0628,0.0910,0.0160,0.1797,0.0921,0.1696,0.1372,0.0000	19.3106	217.578

ag2_det(M,u,100,ni,0.25,0.01)

ni	Cromosoma: u = (3,7,9,16,17,27,30,34,46,48)	Aptitud	Tiempo (s)
480	0.1708,0.0721,0.0736,0.0656,0.0161,0.2139,0.0604,0.1927,0.1348,0.0000	19.5651	236.922
430	0.1725,0.0729,0.0796,0.0644,0.0130,0.2117,0.0591,0.1939,0.1329,0.0000	19.5643	212.671
250	0.1758,0.0718,0.0701,0.0670,0.0134,0.2118,0.0583,0.1986,0.1331,0.0001	19.5614	123.640
370	0.1671,0.0720,0.0729,0.0616,0.0274,0.2124,0.0588,0.1941,0.1337,0.0000	19.5601	182.687
390	0.1696,0.0729,0.0711,0.0657,0.0264,0.2055,0.0591,0.1939,0.1358,0.0000	19.5592	193.235
400	0.1693,0.0723,0.0780,0.0640,0.0271,0.2065,0.0584,0.1933,0.1311,0.0000	19.5591	198.047
290	0.1670,0.0753,0.0810,0.0605,0.0187,0.2154,0.0535,0.1928,0.1358,0.0000	19.5588	143.468
450	0.1685,0.0780,0.0712,0.0643,0.0259,0.2037,0.0592,0.1957,0.1335,0.0000	19.5578	222.766
490	0.1813,0.0744,0.0604,0.0653,0.0190,0.2115,0.0595,0.1964,0.1322,0.0000	19.5569	241.907
500	0.1764,0.0637,0.0797,0.0653,0.0131,0.2193,0.0598,0.1951,0.1274,0.0002	19.5567	247.188
360	0.1717,0.0735,0.0631,0.0638,0.0185,0.2282,0.0556,0.1951,0.1305,0.0000	19.5564	177.860
330	0.1675,0.0741,0.0716,0.0566,0.0219,0.2120,0.0565,0.1983,0.1415,0.0000	19.5548	162.703
470	0.1793,0.0736,0.0731,0.0674,0.0191,0.2093,0.0638,0.1949,0.1195,0.0000	19.5535	232.218
440	0.1738,0.0660,0.0662,0.0669,0.0137,0.2197,0.0664,0.1960,0.1313,0.0000	19.5530	217.484
150	0.1768,0.0586,0.0785,0.0622,0.0139,0.2198,0.0586,0.1965,0.1351,0.0000	19.5518	74.156
350	0.1710,0.0769,0.0611,0.0664,0.0306,0.2178,0.0577,0.1962,0.1223,0.0000	19.5499	172.750

130	0.1674,0.0847,0.0773,0.0646,0.0193,0.2021,0.0579,0.1935,0.1329,0.0003	19.5489	64.344
310	0.1732,0.0734,0.0759,0.0652,0.0170,0.2081,0.0688,0.1868,0.1316,0.0000	19.5487	153.157
160	0.1717,0.0692,0.0845,0.0640,0.0062,0.2224,0.0569,0.1945,0.1299,0.0007	19.5479	79.078
280	0.1688,0.0656,0.0772,0.0643,0.0173,0.2077,0.0692,0.1924,0.1375,0.0000	19.5471	138.297
270	0.1644,0.0761,0.0775,0.0603,0.0203,0.1993,0.0658,0.1954,0.1409,0.0000	19.5465	133.453
320	0.1717,0.0747,0.0538,0.0692,0.0270,0.2145,0.0552,0.1974,0.1365,0.0000	19.5464	158.000
420	0.1715,0.0738,0.0625,0.0545,0.0221,0.2169,0.0575,0.2036,0.1376,0.0000	19.5455	208.078
260	0.1682,0.0756,0.0738,0.0610,0.0249,0.1957,0.0609,0.1899,0.1500,0.0000	19.5434	128.468
300	0.1726,0.0837,0.0837,0.0642,0.0153,0.2179,0.0573,0.1937,0.1115,0.0001	19.5417	148.203
240	0.1685,0.0675,0.0643,0.0699,0.0343,0.2088,0.0561,0.1943,0.1357,0.0006	19.5380	118.515
340	0.1616,0.0757,0.1011,0.0629,0.0158,0.2018,0.0580,0.1901,0.1330,0.0000	19.5379	167.875
200	0.1787,0.0728,0.0698,0.0676,0.0331,0.2148,0.0543,0.1969,0.1120,0.0000	19.5375	98.844
460	0.1690,0.0762,0.0977,0.0607,0.0183,0.1950,0.0607,0.1893,0.1331,0.0000	19.5365	227.531
180	0.1815,0.0710,0.0861,0.0686,0.0162,0.1908,0.0587,0.1955,0.1314,0.0002	19.5363	89.047
380	0.1689,0.0708,0.0967,0.0644,0.0242,0.1928,0.0588,0.1899,0.1335,0.0000	19.5351	188.187
410	0.1867,0.0845,0.0590,0.0633,0.0180,0.2007,0.0567,0.1993,0.1318,0.0000	19.5346	203.156
230	0.1646,0.0766,0.0976,0.0683,0.0140,0.1923,0.0626,0.1903,0.1335,0.0002	19.5299	113.641
140	0.1672,0.0811,0.0872,0.0701,0.0193,0.1950,0.0573,0.2001,0.1225,0.0002	19.5293	69.375
190	0.1722,0.0709,0.0994,0.0748,0.0110,0.1973,0.0589,0.1879,0.1272,0.0004	19.5234	93.875
210	0.1680,0.0717,0.0958,0.0718,0.0129,0.1874,0.0661,0.1880,0.1383,0.0000	19.5232	103.875
220	0.1922,0.0796,0.0585,0.0701,0.0176,0.2204,0.0529,0.2004,0.1083,0.0000	19.5211	108.860
170	0.1701,0.0810,0.0930,0.0613,0.0232,0.1838,0.0666,0.1829,0.1380,0.0001	19.5111	84.063
120	0.1845,0.0700,0.0551,0.0546,0.0466,0.1976,0.0633,0.1972,0.1306,0.0005	19.4874	59.422
80	0.1645,0.0795,0.0682,0.0611,0.0187,0.1908,0.0747,0.1849,0.1570,0.0006	19.4871	39.719
50	0.1632,0.0592,0.0738,0.0688,0.0170,0.2310,0.0511,0.1828,0.1517,0.0014	19.4865	24.953
90	0.1641,0.0991,0.0764,0.0623,0.0175,0.1880,0.0644,0.1819,0.1459,0.0004	19.4862	44.718
100	0.1811,0.0921,0.0792,0.0689,0.0256,0.1970,0.0593,0.1852,0.1098,0.0018	19.4835	49.750
60	0.2018,0.0769,0.0644,0.0646,0.0450,0.1822,0.0557,0.1971,0.1113,0.0010	19.4414	30.000
110	0.1815,0.0953,0.1089,0.0462,0.0111,0.1972,0.0647,0.1891,0.1050,0.0010	19.4086	54.547
70	0.1624,0.0996,0.0720,0.0479,0.0352,0.1779,0.0706,0.1881,0.1439,0.0024	19.4075	34.781
30	0.1577,0.0584,0.0491,0.0812,0.0298,0.2116,0.0517,0.2399,0.1192,0.0014	19.3207	15.110
40	0.2048,0.1041,0.0652,0.0231,0.0130,0.2208,0.0555,0.1914,0.1153,0.0068	19.2212	20.078
20	0.1574,0.0891,0.0685,0.0582,0.0422,0.1890,0.0643,0.1684,0.1436,0.0193	19.1955	10.203
10	0.1503,0.1434,0.0605,0.1284,0.0475,0.1049,0.0402,0.1758,0.1438,0.0052	18.4985	5.734

ag2_det(M,u,100,100,pc,0.01)

pc	Cromosoma: u = (3,7,9,16,17,27,30,34,46,48)	Aptitud	Tiempo (s)
0.07	0.1727,0.0717,0.0806,0.0640,0.0150,0.2132,0.0609,0.1956,0.1260,0.0003	19.5580	49.531
0.3	0.1720,0.0796,0.0841,0.0636,0.0176,0.1974,0.0613,0.1893,0.1351,0.0000	19.5504	49.719
0.24	0.1648,0.0723,0.0885,0.0688,0.0161,0.2041,0.0573,0.1890,0.1381,0.0010	19.5399	49.546
0.16	0.1800,0.0717,0.0569,0.0604,0.0219,0.2169,0.0553,0.1910,0.1458,0.0001	19.5383	49.500
0.18	0.1748,0.0723,0.0752,0.0710,0.0075,0.2052,0.0659,0.1958,0.1313,0.0010	19.5376	49.516
0.1	0.1710,0.0754,0.0784,0.0677,0.0301,0.1969,0.0575,0.1883,0.1339,0.0008	19.5369	49.422
0.48	0.1761,0.0843,0.0774,0.0679,0.0026,0.2052,0.0662,0.1914,0.1288,0.0001	19.5368	49.906
0.4	0.1718,0.0771,0.0829,0.0733,0.0177,0.1944,0.0590,0.1877,0.1356,0.0005	19.5352	49.812
0.35	0.1751,0.0729,0.0624,0.0528,0.0268,0.2108,0.0650,0.2015,0.1327,0.0000	19.5338	49.750
0.23	0.1762,0.0761,0.0752,0.0664,0.0245,0.2004,0.0506,0.2027,0.1274,0.0005	19.5332	49.578
0.12	0.1722,0.0698,0.0927,0.0630,0.0108,0.1961,0.0575,0.1868,0.1509,0.0002	19.5306	49.469

0.02	0.1760,0.0729,0.0815,0.0552,0.0096,0.2122,0.0533,0.2080,0.1309,0.0004	19.5267	49.391
0.26	0.1715,0.0638,0.0645,0.0560,0.0187,0.2081,0.0578,0.2001,0.1594,0.0001	19.5248	49.625
0.45	0.1779,0.0672,0.0580,0.0746,0.0115,0.2216,0.0560,0.1893,0.1432,0.0007	19.5221	49.828
0.37	0.1769,0.0818,0.0864,0.0526,0.0031,0.2060,0.0587,0.1910,0.1427,0.0008	19.5185	49.703
0.43	0.1688,0.0737,0.0509,0.0630,0.0430,0.2174,0.0552,0.2044,0.1235,0.0001	19.5184	49.797
0.04	0.1858,0.0791,0.0869,0.0680,0.0189,0.2015,0.0521,0.1979,0.1096,0.0002	19.5163	49.531
0.22	0.1756,0.0818,0.0526,0.0713,0.0286,0.2219,0.0558,0.2039,0.1081,0.0004	19.5144	49.609
0.19	0.1796,0.0712,0.0969,0.0651,0.0198,0.1935,0.0544,0.1918,0.1263,0.0014	19.5098	49.672
0.31	0.1619,0.0840,0.0956,0.0645,0.0165,0.1836,0.0676,0.1881,0.1379,0.0003	19.5054	49.718
0.33	0.1690,0.0782,0.1021,0.0779,0.0131,0.1850,0.0608,0.1876,0.1263,0.0000	19.4997	49.656
0.41	0.1798,0.0763,0.0833,0.0562,0.0213,0.1981,0.0559,0.1960,0.1297,0.0034	19.4973	49.812
0.11	0.1748,0.0932,0.0564,0.0637,0.0291,0.2240,0.0581,0.1889,0.1106,0.0012	19.4969	49.469
0.46	0.1723,0.0712,0.0858,0.0760,0.0267,0.1776,0.0520,0.1890,0.1494,0.0000	19.4934	49.766
0.47	0.1707,0.0519,0.0758,0.0714,0.0330,0.2033,0.0534,0.2008,0.1377,0.0020	19.4916	49.844
0.01	0.1574,0.0773,0.0869,0.0660,0.0263,0.1721,0.0595,0.1962,0.1574,0.0009	19.4784	49.421
0.34	0.1668,0.1017,0.0934,0.0638,0.0212,0.1969,0.0587,0.1867,0.1104,0.0004	19.4760	49.734
0.5	0.1734,0.0786,0.0467,0.0654,0.0414,0.1932,0.0631,0.1857,0.1513,0.0012	19.4752	49.765
0.32	0.1966,0.0686,0.0739,0.0580,0.0080,0.2121,0.0493,0.1986,0.1312,0.0037	19.4713	49.703
0.17	0.1567,0.0734,0.1057,0.0676,0.0165,0.2036,0.0546,0.1787,0.1398,0.0034	19.4693	49.547
0.39	0.1464,0.0686,0.1119,0.0654,0.0070,0.2172,0.0554,0.1919,0.1336,0.0026	19.4693	49.688
0.03	0.1676,0.1053,0.0769,0.0488,0.0169,0.1982,0.0645,0.1868,0.1341,0.0009	19.4625	49.453
0.05	0.1915,0.0847,0.0445,0.0550,0.0211,0.2001,0.0518,0.2039,0.1451,0.0023	19.4598	49.453
0.44	0.1744,0.0713,0.0378,0.0749,0.0269,0.1975,0.0654,0.2016,0.1468,0.0034	19.4555	49.719
0.14	0.1403,0.0739,0.1134,0.0626,0.0321,0.1954,0.0614,0.1855,0.1339,0.0015	19.4523	49.515
0.36	0.1646,0.0628,0.0762,0.0770,0.0332,0.1881,0.0460,0.1955,0.1539,0.0027	19.4479	49.797
0.42	0.1482,0.0622,0.0822,0.0772,0.0295,0.1966,0.0605,0.1786,0.1632,0.0018	19.4466	49.765
0.2	0.1652,0.0507,0.0694,0.0795,0.0530,0.2101,0.0566,0.1817,0.1338,0.0000	19.4408	49.547
0.15	0.1721,0.0781,0.0489,0.0749,0.0496,0.1686,0.0651,0.1937,0.1490,0.0000	19.4406	49.532
0.21	0.1715,0.0953,0.0942,0.0634,0.0065,0.1827,0.0621,0.1831,0.1373,0.0039	19.4382	49.687
0.28	0.1623,0.0967,0.0747,0.0620,0.0232,0.1899,0.0501,0.2003,0.1354,0.0054	19.4340	49.594
0.09	0.1447,0.0904,0.1000,0.0660,0.0230,0.1778,0.0545,0.1858,0.1557,0.0021	19.4338	49.469
0.25	0.1531,0.0829,0.0982,0.0902,0.0120,0.1792,0.0677,0.1780,0.1386,0.0001	19.4216	49.656
0.38	0.1656,0.0693,0.0818,0.0676,0.0566,0.1617,0.0625,0.1854,0.1488,0.0007	19.4159	49.704
0.13	0.1803,0.1009,0.0738,0.0700,0.0147,0.1927,0.0770,0.1906,0.0988,0.0012	19.4125	49.468
0.08	0.1799,0.0560,0.0898,0.0436,0.0199,0.1988,0.0592,0.1898,0.1577,0.0053	19.3987	49.640
0.49	0.1659,0.0693,0.0666,0.1047,0.0193,0.1909,0.0660,0.1979,0.1194,0.0000	19.3976	49.859
0.06	0.1480,0.0570,0.1357,0.0699,0.0202,0.1879,0.0555,0.1889,0.1353,0.0016	19.3916	49.500
0.29	0.1828,0.0521,0.1211,0.0753,0.0192,0.2122,0.0521,0.1881,0.0954,0.0017	19.3904	49.547
0.27	0.1625,0.0859,0.0711,0.0390,0.0161,0.2123,0.0698,0.1885,0.1472,0.0076	19.3720	49.640

ag2_det(M,u,100,100,0.25,pm)

pm	Cromosoma: u = (3,7,9,16,17,27,30,34,46,48)	Aptitud	Tiempo (s)
0.02	0.1716,0.0652,0.1073,0.0754,0.0127,0.1968,0.0558,0.1836,0.1316,0.0000	19.5071	49.563
0.025	0.1775,0.0731,0.0682,0.0752,0.0082,0.2354,0.0534,0.1963,0.1108,0.0019	19.5022	49.547
0.01	0.1608,0.0564,0.0791,0.0495,0.0341,0.2170,0.0564,0.1958,0.1509,0.0000	19.5014	49.469
0.03	0.1788,0.0653,0.0779,0.0808,0.0248,0.2019,0.0625,0.1757,0.1323,0.0000	19.5007	49.609
0.045	0.1763,0.0743,0.1048,0.0637,0.0028,0.1957,0.0701,0.1902,0.1216,0.0005	19.4953	49.672
0.04	0.1684,0.0720,0.1000,0.0594,0.0159,0.2146,0.0491,0.1858,0.1313,0.0035	19.4823	49.562

0.055	0.1583,0.0799,0.0760,0.0583,0.0111,0.2404,0.0414,0.1958,0.1366,0.0022	19.4706	49.531
0.005	0.1657,0.0899,0.0812,0.0637,0.0400,0.1737,0.0595,0.1894,0.1354,0.0015	19.4691	49.521
0.035	0.1732,0.0634,0.1199,0.0579,0.0065,0.2021,0.0482,0.1899,0.1380,0.0009	19.4621	49.656
0.06	0.1696,0.0905,0.0685,0.0710,0.0342,0.1665,0.0667,0.1888,0.1434,0.0008	19.4577	49.672
0.08	0.1523,0.0686,0.0868,0.0520,0.0445,0.2288,0.0374,0.2005,0.1291,0.0000	19.4490	49.625
0.07	0.1766,0.0632,0.0829,0.0694,0.0074,0.1934,0.0684,0.1906,0.1413,0.0068	19.4453	49.704
0.075	0.1608,0.0676,0.0608,0.0608,0.0638,0.2063,0.0536,0.1974,0.1261,0.0028	19.4294	49.703
0.085	0.1599,0.0677,0.1036,0.0765,0.0190,0.2054,0.0640,0.1874,0.1106,0.0059	19.4216	49.734
0.095	0.1794,0.0798,0.1102,0.0569,0.0292,0.2175,0.0416,0.1769,0.1081,0.0004	19.4212	49.750
0.135	0.1541,0.0815,0.0541,0.0600,0.0545,0.2064,0.0597,0.2173,0.1115,0.0009	19.4155	49.703
0.05	0.1904,0.0821,0.0725,0.0609,0.0141,0.1836,0.0644,0.1918,0.1318,0.0084	19.4103	49.641
0.065	0.1678,0.0838,0.1202,0.0554,0.0020,0.1994,0.0507,0.1956,0.1198,0.0053	19.3890	49.703
0.1	0.2003,0.0675,0.0574,0.0599,0.0398,0.1574,0.0622,0.2109,0.1443,0.0003	19.3890	49.750
0.125	0.1467,0.0564,0.1175,0.0804,0.0440,0.1869,0.0652,0.1821,0.1205,0.0003	19.3808	49.890
0.015	0.1620,0.0722,0.1079,0.0571,0.0560,0.1643,0.0553,0.1869,0.1366,0.0017	19.3755	49.672
0.15	0.1770,0.0894,0.0798,0.0405,0.0290,0.1803,0.0519,0.1763,0.1752,0.0006	19.3675	49.953
0.11	0.1769,0.0785,0.0623,0.0682,0.0461,0.1672,0.0812,0.1763,0.1407,0.0026	19.3580	49.873
0.145	0.1940,0.1086,0.0856,0.0535,0.0100,0.1613,0.0678,0.1887,0.1269,0.0036]	19.3256	49.782
0.105	0.1279,0.0899,0.1362,0.0682,0.0078,0.2094,0.0551,0.2017,0.1036,0.0002	19.3132	49.844
0.12	0.1838,0.1161,0.0618,0.0735,0.0257,0.1766,0.0475,0.1709,0.1423,0.0018	19.3038	49.797
0.19	0.1634,0.0806,0.0848,0.0730,0.0395,0.1500,0.0435,0.1880,0.1733,0.0039	19.3008	49.985
0.09	0.1955,0.0861,0.0522,0.0970,0.0306,0.2015,0.0731,0.1756,0.0856,0.0028	19.2838	49.766
0.165	0.1283,0.1066,0.0781,0.0775,0.0367,0.2015,0.0525,0.1686,0.1450,0.0052	19.2804	49.890
0.185	0.1751,0.1026,0.0718,0.0880,0.0405,0.1685,0.0744,0.1839,0.0920,0.0032	19.2753	49.921
0.14	0.1608,0.0491,0.0940,0.0618,0.0638,0.1826,0.0812,0.1877,0.1167,0.0023	19.2627	49.859
0.13	0.1879,0.0986,0.0938,0.0614,0.0078,0.1621,0.0613,0.2018,0.1127,0.0126	19.2412	49.860
0.115	0.2232,0.0784,0.0537,0.0814,0.0117,0.2181,0.0422,0.2278,0.0634,0.0001	19.2386	49.766
0.155	0.1563,0.0329,0.0885,0.0480,0.0651,0.1944,0.0737,0.1898,0.1494,0.0019	19.2369	50.000
0.195	0.1417,0.0880,0.1072,0.0786,0.0034,0.1946,0.0428,0.2313,0.1102,0.0022	19.2301	49.875
0.17	0.1982,0.0519,0.0387,0.0994,0.0013,0.2241,0.0685,0.1965,0.1095,0.0119	19.2275	49.906
0.21	0.1558,0.0974,0.1498,0.0708,0.0040,0.1451,0.0649,0.1802,0.1311,0.0009	19.2224	50.000
0.18	0.1736,0.1123,0.0931,0.0522,0.0243,0.1667,0.0772,0.1771,0.1140,0.0095	19.2146	49.953
0.2	0.1484,0.1148,0.0194,0.0638,0.0016,0.2019,0.0491,0.2104,0.1903,0.0003	19.1809	50.063
0.205	0.1640,0.0524,0.1047,0.0709,0.0544,0.1334,0.0433,0.2261,0.1501,0.0007	19.1494	50.063
0.16	0.1119,0.0743,0.1579,0.0813,0.0046,0.1738,0.0695,0.1689,0.1562,0.0016	19.1448	49.860
0.215	0.1674,0.1053,0.0377,0.0805,0.0526,0.1589,0.0576,0.1655,0.1661,0.0084	19.1228	50.000
0.23	0.1936,0.0781,0.0036,0.0922,0.0397,0.1787,0.0777,0.2059,0.1147,0.0158	19.0895	50.078
0.24	0.1916,0.0765,0.0231,0.1139,0.0372,0.1892,0.0647,0.2000,0.0917,0.0121	19.0800	50.078
0.22	0.1668,0.1168,0.0826,0.0679,0.0735,0.1939,0.0166,0.1864,0.0941,0.0014	19.0655	50.031
0.235	0.1672,0.0760,0.0497,0.0764,0.0112,0.1882,0.0937,0.1618,0.1594,0.0164	19.0650	50.031
0.245	0.1785,0.0711,0.0659,0.0925,0.0553,0.1330,0.0828,0.1982,0.1107,0.0120	19.0580	50.141
0.25	0.1602,0.0485,0.0904,0.0580,0.0350,0.1849,0.0836,0.1811,0.1318,0.0265	19.0190	50.125
0.175	0.1421,0.0475,0.1377,0.0796,0.0338,0.1525,0.0958,0.1533,0.1539,0.0038	18.9799	49.859
0.225	0.1849,0.0921,0.1043,0.0881,0.0189,0.1072,0.0628,0.1713,0.1552,0.0152	18.9646	50.047

Algoritmo genético 5: ag5_iar(0.0031,0.0073,0.0002,100,nc,ni,pc,pm)

ag5_iar(0.0031,0.0073,0.0002,100,nc,100,0.25,0.01)

nc	Cromosoma	Aptitud	Tiempo (s)
20	0.54422266984069	0.000989078623903056	1.391
50	0.54422266984069	0.000989078623903056	4.407
60	0.54422266984069	0.000989078623903056	5.672
70	0.54422266984069	0.000989078623903056	7.062
80	0.54422266984069	0.000989078623903056	8.625
90	0.54422266984069	0.000989078623903056	10.359
100	0.54422266984069	0.000989078623903056	12.203
110	0.54422266984069	0.000989078623903056	14.219
120	0.54422266984069	0.000989078623903056	16.297
130	0.54422266984069	0.000989078623903056	18.516
140	0.54422266984069	0.000989078623903056	20.937
150	0.54422266984069	0.000989078623903056	23.516
160	0.54422266984069	0.000989078623903056	26.109
170	0.54422266984069	0.000989078623903056	28.922
180	0.54422266984069	0.000989078623903056	31.906
190	0.54422266984069	0.000989078623903056	35.094
200	0.54422266984069	0.000989078623903056	38.437
210	0.54422266984069	0.000989078623903056	41.750
220	0.54422266984069	0.000989078623903056	45.329
230	0.54422266984069	0.000989078623903056	48.766
240	0.54422266984069	0.000989078623903056	52.610
250	0.54422266984069	0.000989078623903056	56.531
260	0.54422266984069	0.000989078623903056	60.688
270	0.54422266984069	0.000989078623903056	64.797
280	0.54422266984069	0.000989078623903056	69.203
290	0.54422266984069	0.000989078623903056	73.937
300	0.54422266984069	0.000989078623903056	78.250
310	0.54422266984069	0.000989078623903056	83.313
320	0.54422266984069	0.000989078623903056	87.735
330	0.54422266984069	0.000989078623903056	92.937
340	0.54422266984069	0.000989078623903056	98.047
350	0.54422266984069	0.000989078623903056	103.578
360	0.54422266984069	0.000989078623903056	108.922
370	0.54422266984069	0.000989078623903056	114.390
380	0.54422266984069	0.000989078623903056	120.235
390	0.54422266984069	0.000989078623903056	126.109
400	0.54422266984069	0.000989078623903056	132.125
410	0.54422266984069	0.000989078623903056	138.343
420	0.54422266984069	0.000989078623903056	144.703
430	0.54422266984069	0.000989078623903056	151.203
440	0.54422266984069	0.000989078623903056	157.719
450	0.54422266984069	0.000989078623903056	164.328
460	0.54422266984069	0.000989078623903056	171.250
470	0.54422266984069	0.000989078623903056	178.188
480	0.54422266984069	0.000989078623903056	185.328
490	0.54422266984069	0.000989078623903056	192.922
500	0.54422266984069	0.000989078623903056	200.125
10	0.54416163095892	0.000989078623901193	0.656
30	0.54416163095892	0.000989078623901193	2.219

40	0.54416163095892	0.000989078623901193	3.219
----	------------------	----------------------	-------

ag5_iar(0.0031,0.0073,0.0002,100,100,ni,0.25,0.01)

ni	Cromosoma	Aptitud	Tiempo (s)
20	0.54422266984069	0.000989078623903056	2.500
30	0.54422266984069	0.000989078623903056	3.828
40	0.54422266984069	0.000989078623903056	4.891
50	0.54422266984069	0.000989078623903056	6.140
60	0.54422266984069	0.000989078623903056	7.391
70	0.54422266984069	0.000989078623903056	8.610
80	0.54422266984069	0.000989078623903056	9.797
90	0.54422266984069	0.000989078623903056	11.047
100	0.54422266984069	0.000989078623903056	12.204
120	0.54422266984069	0.000989078623903056	14.578
130	0.54422266984069	0.000989078623903056	15.797
140	0.54422266984069	0.000989078623903056	17.110
150	0.54422266984069	0.000989078623903056	18.250
160	0.54422266984069	0.000989078623903056	19.422
170	0.54422266984069	0.000989078623903056	20.766
180	0.54422266984069	0.000989078623903056	21.890
190	0.54422266984069	0.000989078623903056	23.078
200	0.54422266984069	0.000989078623903056	24.250
210	0.54422266984069	0.000989078623903056	25.578
220	0.54422266984069	0.000989078623903056	26.844
230	0.54422266984069	0.000989078623903056	27.922
240	0.54422266984069	0.000989078623903056	29.235
250	0.54422266984069	0.000989078623903056	30.360
260	0.54422266984069	0.000989078623903056	31.563
270	0.54422266984069	0.000989078623903056	32.812
280	0.54422266984069	0.000989078623903056	34.031
290	0.54422266984069	0.000989078623903056	35.250
300	0.54422266984069	0.000989078623903056	36.453
310	0.54422266984069	0.000989078623903056	37.641
320	0.54422266984069	0.000989078623903056	38.797
330	0.54422266984069	0.000989078623903056	39.984
340	0.54422266984069	0.000989078623903056	41.187
350	0.54422266984069	0.000989078623903056	42.391
360	0.54422266984069	0.000989078623903056	43.875
370	0.54422266984069	0.000989078623903056	45.219
380	0.54422266984069	0.000989078623903056	46.047
390	0.54422266984069	0.000989078623903056	47.281
400	0.54422266984069	0.000989078623903056	48.657
410	0.54422266984069	0.000989078623903056	49.703
420	0.54422266984069	0.000989078623903056	50.922
430	0.54422266984069	0.000989078623903056	52.281
440	0.54422266984069	0.000989078623903056	53.500
450	0.54422266984069	0.000989078623903056	54.734
460	0.54422266984069	0.000989078623903056	55.906

470	0.54422266984069	0.000989078623903056	56.953
480	0.54422266984069	0.000989078623903056	58.110
490	0.54422266984069	0.000989078623903056	59.406
500	0.54422266984069	0.000989078623903056	60.750
110	0.54416163095892	0.000989078623901193	13.391
10	0.54410059207715	0.000989078604044835	1.281

ag5_iar(0.0031,0.0073,0.0002,100,100,100,pc,0.01)

pc	Cromosoma	Aptitud	Tiempo (s)
0.01	0.54422266984069	0.000989078623903056	12.172
0.02	0.54422266984069	0.000989078623903056	12.110
0.03	0.54422266984069	0.000989078623903056	12.204
0.04	0.54422266984069	0.000989078623903056	12.140
0.05	0.54422266984069	0.000989078623903056	12.156
0.07	0.54422266984069	0.000989078623903056	12.109
0.08	0.54422266984069	0.000989078623903056	12.156
0.09	0.54422266984069	0.000989078623903056	12.203
0.11	0.54422266984069	0.000989078623903056	12.141
0.12	0.54422266984069	0.000989078623903056	12.141
0.13	0.54422266984069	0.000989078623903056	12.141
0.14	0.54422266984069	0.000989078623903056	12.172
0.15	0.54422266984069	0.000989078623903056	12.172
0.16	0.54422266984069	0.000989078623903056	12.188
0.17	0.54422266984069	0.000989078623903056	12.141
0.18	0.54422266984069	0.000989078623903056	12.140
0.19	0.54422266984069	0.000989078623903056	12.188
0.2	0.54422266984069	0.000989078623903056	12.219
0.22	0.54422266984069	0.000989078623903056	12.188
0.23	0.54422266984069	0.000989078623903056	12.172
0.24	0.54422266984069	0.000989078623903056	12.156
0.25	0.54422266984069	0.000989078623903056	12.187
0.27	0.54422266984069	0.000989078623903056	12.250
0.28	0.54422266984069	0.000989078623903056	12.203
0.29	0.54422266984069	0.000989078623903056	12.187
0.3	0.54422266984069	0.000989078623903056	12.250
0.31	0.54422266984069	0.000989078623903056	12.219
0.32	0.54422266984069	0.000989078623903056	12.188
0.33	0.54422266984069	0.000989078623903056	12.235
0.34	0.54422266984069	0.000989078623903056	12.250
0.35	0.54422266984069	0.000989078623903056	12.219
0.37	0.54422266984069	0.000989078623903056	12.235
0.38	0.54422266984069	0.000989078623903056	12.250
0.39	0.54422266984069	0.000989078623903056	12.234
0.4	0.54422266984069	0.000989078623903056	12.235
0.41	0.54422266984069	0.000989078623903056	12.266
0.42	0.54422266984069	0.000989078623903056	12.234
0.43	0.54422266984069	0.000989078623903056	12.313
0.44	0.54422266984069	0.000989078623903056	12.281

0.45	0.54422266984069	0.000989078623903056	12.250
0.46	0.54422266984069	0.000989078623903056	12.281
0.47	0.54422266984069	0.000989078623903056	12.250
0.48	0.54422266984069	0.000989078623903056	12.297
0.49	0.54422266984069	0.000989078623903056	12.265
0.5	0.54422266984069	0.000989078623903056	12.313
0.06	0.54416163095892	0.000989078623901193	12.063
0.1	0.54416163095892	0.000989078623901193	12.109
0.21	0.54416163095892	0.000989078623901193	12.234
0.26	0.54416163095892	0.000989078623901193	12.172
0.36	0.54416163095892	0.000989078623901193	12.235

ag5_iar(0.0031,0.0073,0.0002,100,100,100,0.25,pm)

pm	Cromosoma	Aptitud	Tiempo (s)
0.005	0.54422266984069	0.000989078623903056	12.250
0.01	0.54422266984069	0.000989078623903056	12.172
0.015	0.54422266984069	0.000989078623903056	12.172
0.02	0.54422266984069	0.000989078623903056	12.203
0.025	0.54422266984069	0.000989078623903056	12.156
0.035	0.54422266984069	0.000989078623903056	12.172
0.04	0.54422266984069	0.000989078623903056	12.188
0.045	0.54422266984069	0.000989078623903056	12.156
0.05	0.54422266984069	0.000989078623903056	12.203
0.055	0.54422266984069	0.000989078623903056	12.203
0.06	0.54422266984069	0.000989078623903056	12.266
0.065	0.54422266984069	0.000989078623903056	12.250
0.07	0.54422266984069	0.000989078623903056	12.250
0.075	0.54422266984069	0.000989078623903056	12.203
0.08	0.54422266984069	0.000989078623903056	12.172
0.085	0.54422266984069	0.000989078623903056	12.203
0.09	0.54422266984069	0.000989078623903056	12.234
0.095	0.54422266984069	0.000989078623903056	12.250
0.1	0.54422266984069	0.000989078623903056	12.219
0.105	0.54422266984069	0.000989078623903056	12.218
0.11	0.54422266984069	0.000989078623903056	12.203
0.115	0.54422266984069	0.000989078623903056	12.250
0.12	0.54422266984069	0.000989078623903056	12.250
0.125	0.54422266984069	0.000989078623903056	12.266
0.13	0.54422266984069	0.000989078623903056	12.234
0.135	0.54422266984069	0.000989078623903056	12.218
0.14	0.54422266984069	0.000989078623903056	12.234
0.145	0.54422266984069	0.000989078623903056	12.297
0.15	0.54422266984069	0.000989078623903056	12.203
0.155	0.54422266984069	0.000989078623903056	12.188
0.16	0.54422266984069	0.000989078623903056	12.266
0.165	0.54422266984069	0.000989078623903056	12.235
0.175	0.54422266984069	0.000989078623903056	12.234
0.18	0.54422266984069	0.000989078623903056	12.235

0.19	0.54422266984069	0.000989078623903056	12.235
0.2	0.54422266984069	0.000989078623903056	12.235
0.205	0.54422266984069	0.000989078623903056	12.250
0.21	0.54422266984069	0.000989078623903056	12.219
0.22	0.54422266984069	0.000989078623903056	12.250
0.225	0.54422266984069	0.000989078623903056	12.203
0.23	0.54422266984069	0.000989078623903056	12.265
0.03	0.54416163095892	0.000989078623901193	12.218
0.17	0.54416163095892	0.000989078623901193	12.250
0.185	0.54416163095892	0.000989078623901193	12.204
0.195	0.54416163095892	0.000989078623901193	12.219
0.235	0.54416163095892	0.000989078623901193	12.218
0.24	0.54416163095892	0.000989078623901193	12.235
0.245	0.54416163095892	0.000989078623901193	12.281
0.25	0.54416163095892	0.000989078623901193	12.234
0.215	0.54428370872246	0.000989078604050423	12.250

BIBLIOGRAFÍA

1. BAGCHI, Tapan P., *Multiobjective Scheduling by Genetic Algorithms*, Estados Unidos de América, Kluwer Academic Publishers, 1999, 358 pp.
2. BAUER, Richard J. Jr., *Genetic Algorithms and Investment Strategies*, Estados Unidos de América, John Wiley & Sons, Inc., 1994, 308 pp.
3. CHAMBERS, Lance., *The Practical Handbook of Genetic Algorithms. Applications*, 2ª ed., Estados Unidos de América, Chapman & Hall / CRC, 2001, 501 pp.
4. CORDÓN, Oscar, et. al., *Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, Singapur, World Scientific, 2001, 462 pp. (Advances in Fuzzy Systems – Applications and Theory, Vol. 19.)
5. FALKENAUER, Emanuel, *Genetic Algorithms and Grouping Problems*, Gran Bretaña, John Wiley & Sons, Inc., 1998, 220 pp.
6. HAUGEN, Robert A., *Modern Investment Theory*, 5ª ed., Estados Unidos de América, Prentice – Hall, 2001, 656 pp.
7. LUENBERGER, David G., *Investment Science*, Estados Unidos de América, Oxford University Press, 1998, 494 pp.
8. MARTÍNEZ Abascal, Eduardo., *Invertir en Bolsa. Conceptos y estrategias*, España, McGraw – Hill / Interamericana de España, 1999, 396 pp.
9. MICHALEWICZ, Zbigniew., *Genetic Algorithms + Data Structures = Evolutions Programs*, 3ª ed., Estados Unidos de América, Springer – Verlag Berlin Heidelberg, 1996, 387 pp.
10. ROSS, Stephen A., et. al., *Finanzas Corporativas*, 5ª ed., México, McGraw – Hill, 2000, 1049 pp.
11. SHAPIRO, Alan C., *Modern Corporate Finance*, Estados Unidos de América, Prentice – Hall, 1992, 1120 pp.